



TAMPEREEN TEKNILLINEN YLIOPISTO

TATU ANTTONEN
MUSIIKKISOITTIMESTA RIIPPUMATON
ALBUMIKUUNTELUN TUNNISTUS
STATSTER-VERKKOPALVELUSSA

Diplomityö

Tarkastaja: Tommi Mikkonen
Tarkastaja ja aihe hyväksytty
tieto- ja sähkötekniikan
osastoneuvoston kokouksessa
9.3.2011

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO

Tietotekniikan koulutusohjelma

ANTTONEN, TATU: Musiikkisoittimesta riippumaton albumikuuntelun tunnistus Statster-verkkopalvelussa

Diplomityö, 69 sivua

Syyskuu 2011

Pääaine: Ohjelmistotuotanto

Tarkastaja: Tommi Mikkonen

Avainsanat: Statster, musiikki, albumi, metatietokanta, MusicBrainz, Last.fm, musiikki-soitin, foobar2000, automatisointi

Ihmiset kuuntelevat paljon musiikkia. Musiikki merkitsee meille paljon ja myös tunteet liittyvät vahvasti musiikkiin. Tästä syystä on mielekästä taltioida kuunnellusta musiikista kaikki tarpeellinen. Tätä silmälläpitäen musiikin ystäville on luotu verkkopalvelu nimeltä Statster. Statster taltioi käyttäjien kuuntelemat musiikkijulkaisut, joita yleisesti nimitetään albumeiksi.

Tämän diplomityön tarkoituksena on selvittää yleisesti käyttökelpoisia tapoja tunnistaa albumikuuntelu automaattisesti. Tärkein edellytys kuunteluntunnistajan yleiskäyttöisyydelle on riippumattomuus soitinohjelmistosta, jolla musiikkia kuunnellaan. Kuunteluntunnistajan tulee olla itsenäisesti toimiva ja käyttöjärjestelmästä riippumaton sovellus, jotta saavutettaisiin suurin mahdollinen musiikkisoittimien joukko.

Soitinriippumaton albumikuuntelun tunnistus koostuu kappale tietojen saamisesta soittimelta ja näiden tietojen perusteella tehtävästä kuuntelun tunnistamisesta. Ensimmäinen vaihe perustuu soittimien toteuttamiin rajapintoihin, joiden avulla saadaan kuunneltavan kappaleen tiedot. Kappaleesta tarvitaan vähintään sen nimi ja artisti. Toinen vaihe albumikuuntelun tunnistuksessa on itse tunnistus. Tunnistuksella pyritään löytämään oikea albumi ja havaitsemaan, milloin se on kuunneltu kokonaan. Kun albumi on todettu kuunnelluksi sen tiedot voidaan lähettää Statsteriin. Apuna tunnistamisessa käytetään ulkoisia, Internetissä toimivia metatietokantoja.

Työtä varten toteutettiin albumikuuntelun tunnistajan prototyyppi, jolla kokeiltiin ulkoisen metatietokannan mahdollisuuksia kuuntelun tunnistamisessa. Metatietokannaksi valittiin verkkopalvelun MusicBrainz tarjoama avoin musiikin metatietokanta, josta on mahdollista hakea tietoa palvelurajapinnan avulla. Albumin tunnistamisessa käytettiin apuna vain kappaleen nimeä ja artistia. Prototyypillä ei kokeiltu albumikuuntelun tunnistamisen ensimmäistä vaihetta eli soittimien ulkoisia rajapintoja ei hyödynnetty. Prototyyppi osoitti, että albumikuuntelun tunnistaminen on mahdollista käyttäen apuna ulkoista metatietokantaa.

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY

Master's Degree Programme in Computer Science

ANTTONEN, TATU: Music player non-dependent album listening recognition for Statster

Master of Science Thesis, 69 pages

September 2011

Major: Software Engineering

Examiner: Professor Tommi Mikkonen

Keywords: Statster, music, album, metadatabase, MusicBrainz, Last.fm, music player, foobar2000, automatization

Music is a vital part of our lives. Music is significant to us and emotions are strongly connected with it. Therefore, it is beneficial to store all the necessary information of the listened music. For this purpose, an Internet service Statster was created for gathering data of music publications listened by its users.

The goal of this thesis is to study commonly usable methods for recognizing album listening automatically. The most important factor for generalized usage of the listening recognizer is independency from the player software used for listening music. Recognizer must be autonomously functioning application which can be used under various operating systems in order to reach the largest possible group of music players.

Music player non-dependent album listening recognition consists of acquiring track information from the player and listening recognition based on the acquired data. The first phase is based on player-made interfaces, which provide information of the listened tracks. At least track and artist names are required. The second phase, after receiving information from the player, is to recognize the album and send its information to Statster after the album has been completely listened. Metadatabases assist in performing the recognition.

For the study, an album listening recognizer prototype was created, and it was tested for the possibility of using an external metadatabase for listening recognition. Open music metadatabase provided by the web service MusicBrainz was chosen for its possibility to search for data with service interface. For album recognition, only track and artist names were used. The prototype was not tested for the first phase of the recognition and consequently external interfaces of the player were not utilized. The prototype indicated that album listening recognition is possible using external metadatabase.

ALKUSANAT

Tämän diplomityön on mahdollistanut hieno verkkopalvelu nimeltä Statster. Statster on projekti, jonka juuret yltävät vuoteen 2007 saakka. Statsterin perustaja on Teemo Tebest. Itse liityin vuonna 2008 Statsterin kehittäjäryhmään, jota tämän jälkeen saattoi jo kutsua todelliseksi ryhmäksi – koska jäseniä oli kaksi. Aloitin Statsterin kehittämisen tekemällä automaattisesti albumikuuntelut lähettävän lisäosan musiikkisoittimelle Amarok 1.4. Tätä ensimmäistä lisäosaa voidaan pitää alkuna tämän työn idean synnylle, sillä Amarokin jälkeen käytin useampia soittimia, joihin myös tein lisäosat, joilla albumikuuntelut sai lähetettyä automaattisesti. Viimeisimmän lisäosan tein vuoden 2010 joulukuussa, mikä sai minut havahtumaan, että voisi-ko olla soitinohjelmistosta riippumaton tapa tunnistaa ja lähettää albumikuuntelut Statsteriin.

Haluan kiittää tämän työn aiheen hyväksyntyttä professoria Tommi Mikkonen, joka toimi minun ohjaajana ja myös työn tarkastajana. Tommin avulla sain työn sujuvasti aloitettua. Hän antoi riittävän vapaat kädet työn tekemiselle, mutta silti ohjasi työtä koko ajan oikeaan suuntaan. Työn toisena ohjaajana toimi Statsterin perustajajäsen diplomi-insinööri Teemo Tebest, jolta sain hyödyllisiä, työtä selkeyttäviä kommentteja, kiitokset niistä hänelle.

Muita tahoja, joita haluan kiittää ovat isä, Heikki ja äiti, Marja-Terttu, ja diplomi-insinööri Markus Vihottula. Äidin kannustus ja isän rahallinen tuki mahdollistivat työn aloittamisen ja tekemisen. Markus puolestaan auttoi minua työn kieliopillisissa asioissa.

Vielä lopuksi suuret kiitokset Anulle. Ilman sitä pientä toivon hiventä, minkä sain sinulta joskus tammikuun lopun kylminä hetkinä, olisi työ jäänyt aloittamatta, ja painisin yhä tyhjän sivun kanssa. Se toivon hiven, kirjoittamisen edetessä, oli usein ainut voimia antava asia, jota ajattelemalla sain työn etenemään.

Tatu Anttonen
16. elokuuta 2011

SISÄLLYS

1. Johdanto	1
2. Soitinriippumaton albumikuuntelun tunnistus	3
2.1 Kappaletiedon saaminen soittimelta	4
2.1.1 Windows	4
2.1.2 MSN Messenger -rajapinta	5
2.1.3 Linux	6
2.1.4 MPRIS-rajapinta	7
2.1.5 Last.fm	8
2.2 Albumin tunnistaminen	9
2.2.1 Esimerkki: Nirvanan albumin Nevermind kuunteleminen	10
2.2.2 Albumin tunnistamisessa käytettävät metatietokannat	10
2.3 Tiedon haun teknologiat Internetin metatietopalveluissa	11
2.3.1 XML	11
2.3.2 HTTP	12
2.3.3 REST	13
2.3.4 XML-RPC	13
3. Avoimet musiikin metatietokannat	14
3.1 Last.fm	14
3.1.1 Palvelurajapinta	15
3.1.2 Esimerkki: albumin kappalelistan hakeminen	16
3.2 MusicBrainz	20
3.2.1 Palvelurajapinta	21
3.2.2 Esimerkki: albumin kappalelistan hakeminen	25
4. Statster	28
4.1 Käytön aloittaminen	28
4.2 Sivuston pääosiot	29
4.3 Kuuntelut-osio	30
4.3.1 Artisti- ja albumisivut	31
4.3.2 Suosituimmat albumit ja artistit	34
4.3.3 Suosikit	36
4.4 Tietomalli	37
5. Albumikuuntelujen lisääminen Statsteriin	39
5.1 Kuuntelun lisääminen verkkopalvelun kautta	39
5.2 Palvelurajapinta	41
5.3 Oheisohjelmat	43
5.3.1 Lisäosien toimintaperiaate	44
5.3.2 Olemassa olevat oheisohjelmat	45

6. Prototyyppitoteutus	46
6.1 Toteutusympäristö	46
6.1.1 WSH Panel Mod -komponentti	47
6.1.2 Käytetyt WSH Panel Mod -komponentin rajapinnat	48
6.2 Toteutus	48
6.2.1 Yleiskuvaus	49
6.2.2 Julkaisujen hakeminen kuunneltavalle kappaleelle	50
6.2.3 Oikean julkaisun valinta	52
6.2.4 Kappalelistan hakeminen julkaisulle	54
6.2.5 Albumikuuntelun tunnistaminen	55
7. Arviointi	57
7.1 Toteutettujen toimintojen arviointi	57
7.1.1 Julkaisujen hakeminen kuunneltavalle kappaleelle	57
7.1.2 Oikean julkaisun valinta	60
7.1.3 Kappalelistan hakeminen ja kuuntelun tunnistus	62
7.2 Tunnetut puutteet	63
7.3 Jatkokehitysideoita	63
7.3.1 Statster	63
7.3.2 Prototyyppi	64
8. Johtopäätökset	66
Lähteet	66

TERMIT JA NIIDEN MÄÄRITELMÄT

Albumi on kappaleista koostuva musiikillinen kokonaisuus. Albumi voi koostua useasta mediasta, jolla kappaleet ovat.

Albumikuuntelu tarkoittaa albumin kuuntelemista alusta loppuun siten, että sen kaikki kappaleet kuunnellaan siinä järjestyksessä, jossa ne medially ovat.

Artisti Artisti on jokin ryhmä tai henkilö, joka tekee musiikkia.

Julkaisu on jokin musiikillinen kokonaisuus, joka on albumin ilmentymä. Tietystä albumista on voitu esimerkiksi julkaista eri variaatioita, kuten bonuskappaleita sisältäviä tai maakohtaisesti hiukan eri kappalekoostumuksella.

Julkaisuvuosi kertoo koska tietystä albumista tehty julkaisu on julkaistu.

Kappale on pienin musiikillinen teos, joka on taltioitu jollekin medialle. Kappaleella on subjektiivisesti tulkittavissa oleva rakenne.

Kuuntelu tarkoittaa tässä työssä albumikuuntelua.

Media on musiikin taltioimisen ja jakamisen väline, joka voi olla esimerkiksi kovalevy, CD-, DVD- tai Vinyyli-levy. Medially voi olla tallennettuna albumi, useita albumeita tai yksittäisiä kappaleita.

1. JOHDANTO

Ihmiset kuuntelevat paljon musiikkia. Musiikki merkitsee meille paljon ja myös tunteet liittyvät vahvasti musiikkiin. Tästä syystä on mielekästä taltioida kuunnellusta musiikista kaikki tarpeellinen. Tätä silmälläpitäen musiikin ystäville on luotu verkkopalvelu nimeltä Statster¹, joka taltioi käyttäjien kuuntelemat musiikkijulkaisut.

Motivaatio Statsterin toteuttamiselle on syntynyt siitä lähtökohdasta, että musiikkia julkaistaan kokonaisuuksina. Tämän vuoksi on luontevaa saada tietoa kuunnelluista musiikkijulkaisuista. Yksittäisen julkaisun kuuntelemista kutsutaan Statsterin ja tämän työn yhteydessä albumikuunteleksi tai lyhyesti vain kuunteleksi. Albumikuuntelellä tarkoitetaan jonkin musiikillisen kokonaisuuden eli julkaisun – studioalbumi, EP-levy, single – kuuntelemista alusta loppuun.

Statsterin pääasiallinen tehtävä on taltioida käyttäjien kuuntelemat musiikkijulkaisut, tämän vuoksi kuuntelujen lisääminen on pyritty tekemään mahdollisimman helpoksi. Albumikuuntelujen lisääminen Statsteriin onnistuu joko verkkopalvelusta tai suoraan käyttäjän soitinohjelmistosta. Kerätyn kuunteluhistoriatiedon pohjalta Statster koostaa monipuolisia tilastoja käyttäjien kuuntelutottumuksista.

Statsteria varten on tehty useita oheisohjelmia, joiden avulla albumikuuntelut on mahdollista tunnistaa ja lähettää palveluun suoraan musiikkisoittimesta. Toteutetut oheisohjelmat ovat soittimien toimintaa laajentavia lisäosia. Näiden lisäosien tarkoitus on mahdollistaa kattava kuuntelutietojen kerääminen helposti, ilman käyttäjän toimia. Lisäosien avulla toteutettujen oheisohjelmien ansiosta on pystytty täysin automatisoimaan albumikuuntelun tunnistaminen ja lähettäminen, mutta tähän tapaan liittyy myös ongelma.

Ongelma toteutettaessa albumikuuntelun tunnistamisen ja lähettämisen tekevä oheisohjelma laajentamalla soitinohjelmistoa piilee siinä, että ihmiset eivät kuuntele musiikkia aina samalla soittimella. Tietty henkilö ei edes välttämättä käytä aina samaa soitinta. Soittimen laajentamiseen liittyy valtava työ, koska erilaisia musiikkisoittimia on paljon ja jokaisessa soittimessa on erilainen ohjelmointirajapinta sekä käytettävät ohjelmointikielet vaihtelevat. On myös mahdollista, että on olemassa soittimia, joita ei ole mahdollista laajentaa lisäosilla. Siksi lisäosiin pohjautuva tapa ei tule kyseeseen, jos halutaan tarjota mahdollisimman monelle käyttäjälle albumikuuntelun tunnistamisen ja lähettämisen tekevä oheisohjelma.

¹<http://statster.info>

Tämän diplomityön tarkoituksena on selvittää yleisesti käyttökelpoisia tapoja tunnistaa albumikuuntelut automaattisesti. Tärkein edellytys kuunteluntunnistajan yleiskäyttöisyydelle on riippumattomuus soitinohjelmistosta. Kuunteluntunnistajan tulee olla itsenäisesti toimiva ja käyttöjärjestelmästä riippumaton sovellus, jotta saavutettaisiin suurin mahdollinen musiikkisoittimien joukko.

Yleiskäyttöisen albumikuuntelun tunnistusohjelmiston toiminta perustuu soittimien tarjoamiin ulkoisiin rajapintoihin ja metatietokantoihin. Metatietokannat ovat jaettu tässä työssä paikallisiin ja ulkoisiin. Paikalliset ovat suoraan käyttäjän musiikkikokoelmasta luotuja ja ulkoiset jonkin muun tahon tarjoamia, usein Internetissä toimivia verkkopalveluita. Paikalliset metatietokannat ovat kuitenkin rajattu tämän diplomityön ulkopuolelle. Työssä tullaan perehtymään tarkemmin vain ulkoisiin, Internetissä toimiviin musiikin metatietokantoihin ja soittimien tarjoamiin rajapintoihin.

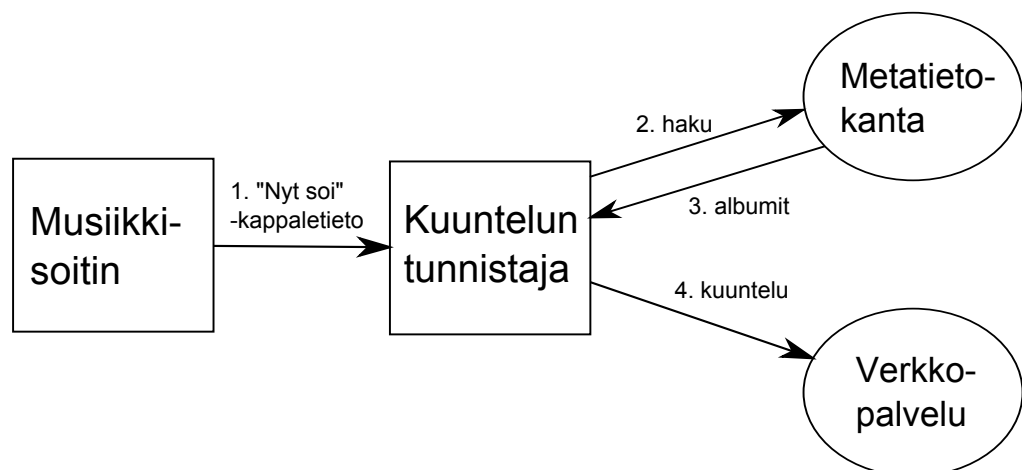
Luvussa 2 tutustutaan työn teoreettiseen taustaan. Luvun rakenne noudattaa soitinriippumattoman albumikuuntelun tunnistuksen päävaiheita. Ensin käsitellään kuunneltavien kappaleiden tietojen saamista soittimilta. Tämän jälkeen tutustutaan albumikuuntelun tunnistamisen teoriaan. Lopuksi tehdään lyhyt katsaus teknologioihin, jotka mahdollistavat verkkopalveluiden ohjelmallisen käytön. Luvussa 3 perehdytään Internetissä toimiviin avoimiin musiikin metatietokantapalveluihin. Tarkasteltavat metatietokannat ovat verkkopalveluiden Last.fm² ja MusicBrainz³ tarjoamat. Luvussa 4 käsitellään Statster-verkkopalvelua. Aluksi tutustutaan verkkopalvelun käyttöön ja rakenteeseen. Tämän jälkeen perehdytään tarkemmin albumikuuntelujen pohjalta luotuihin tilastoihin, ja lopuksi lopuksi näytetään palvelun tietomalli. Luvussa 5 käsitellään sitä, miten albumikuuntelut on mahdollista lisätä Statster-verkkopalveluun. Aluksi perehdytään verkkoselaimen avulla tapahtuvaan kuuntelun lisäämiseen. Tämän jälkeen esitellään Statsterin palvelurajapintaa, joka mahdollistaa kuuntelujen lisäämisen suoraan soitinohjelmistosta. Lopuksi tutustutaan palvelurajapintaa hyödyntäviin oheisohjelmiin eli lisäosiin. Luvussa 6 tutustutaan prototyyppitoteutukseen, jolla kokeillaan albumikuuntelun tunnistamisen teoriaa käytännössä. Prototyypillä testataan vain albumikuuntelun tunnistusta, ei kappaleiden tietojen saamista soittimilta. Käytettävä metatietokanta on MusicBrainz. Luvussa 7 arvioidaan prototyyppitoteutusta ja sen kykyä tunnistaa kuunneltavia albumeita. Soittimien tarjoamat rajapinnat jätetään arvioinnin ulkopuolelle, koska niitä ei toteutettu prototyyppiin. Lopuksi luvussa 8 esitetään tärkeimmät työn esiintuomat johtopäätökset.

²<http://www.last.fm/>

³<http://musicbrainz.org/>

2. SOITINRIIPPUMATON ALBUMIKUUNTELUN TUNNISTUS

Verkkopalvelun, joka kerää käyttäjien albumikuunteluita on tarjottava yleisesti käytökelpoinen tapa kuunteluiden automaattiseen tunnistamiseen ja lähettämiseen. Musiikkisoittimien laajentamiseen pohjautuva tapa ei ole vartenotettava vaihtoehto, kun halutaan tarjota kuunteluntunnistaja mahdollisimman monelle käyttäjälle. Tästä syystä on tärkeää löytää käytettävästä soitinohjelmistosta riippumaton keino albumikuuntelun tunnistamiseen. Kuvassa 2.1 on esitetty soitinriippumattoman albumikuuntelun tunnistamisen peruseriaate.



Kuva 2.1: Soittimesta riippumattoman albumikuuntelun tunnistamisen päävaiheet.

Tämän luvun rakenne noudattaa soitinriippumattoman albumikuuntelun tunnistamisen kahta tärkeintä vaihetta, jotka käyvät ilmi yllä olevasta kuvasta 2.1. Ensimmäinen vaihe on soivan kappaleen tietojen saaminen soittimelta kuuntelun tunnistajalle (kuvassa kohta 1). Tätä käsitellään tämän luvun kohdassa 2.1. Toinen vaihe kuunteluntunnistamisessa on löytää kuunneltujen kappaleiden perusteella albumi, jolta kappaleet ovat (kuvassa kohdat 2 ja 3). Albumin tunnistamisessa apuna käytetään metatietokantoja, joita käsitellään tämän luvun kohdassa 2.2. Albumin kuunnelluksi tunnistamisen jälkeen albumin tiedot lähetetään. Tämä vaihe on nähtävillä kuvan kohdassa 4.

2.1 Kappaletiedon saaminen soittimelta

Soitinohjelmistosta riippumaton albumikuuntelun tunnistus pohjautuu yksinkertaisiin rajapintoihin, joiden avulla soittimet pystyvät jakamaan tietoa kuunneltavasta kappaleesta muille sovelluksille. Yksi tällainen rajapinta on Windowsissa toimivan Live Messenger -pikaviestimen¹ tarjoama ”Nyt soi” -rajapinta, josta käytetään tässä työssä nimeä MSN Messenger -rajapinta. Tämä rajapinta mahdollistaa kuunneltavan kappaleen tietojen näyttämisen tuttaville. Suppeimmillaan ”Nyt soi” -tieto sisältää kappaleen nimen ja artistin. Tiedon määrä riippuu siitä, miten laajasti soitin on toteuttanut rajapinnan, esimerkiksi Spotify lähettää vain kappaleen nimen ja artistin. Usein saatavilla on myös nimi albumista, jolta kappaletta kuunnellaan.

Messenger-pikaviestinverkkoa ei tarvitse välttämättä käyttää Microsoftin Live Messenger -asiakasohjelmalla, sitä voi käyttää esimerkiksi avoimeen lähdekoodiin perustuvalla pikaviestinohjelmalla nimeltä Pidgin². Pidginiin on saatavilla lisäosa nimeltä Music Tracker³, joka pystyy vastaanottamaan ”Nyt soi” -tietoa lähestulkoon kaikilta soittimilta sekä Windows- että Linux-käyttöjärjestelmissä (Turney, s.a.). Windowsissa ”Nyt soi” -tiedon saaminen perustuu pääasiassa soittimien toteuttamaan MSN Messenger -rajapintaan ja Linuxissa puolestaan MPRIS-määrittelyyn (Media Player Remote Interfacing Specification), joka pohjautuu D-Bus-rajapintaan. Myös Last.fm-palvelua hyödyntäen kummassakin käyttöjärjestelmässä on mahdollista saada ”Nyt soi” -tietoa, edellyttäen että käyttäjä lähettää kuuntelemaisensa kappaleet Last.fm:ään.

2.1.1 Windows

Windowsissa lähestulkoon kaikilta soittimilta on mahdollista saada ”Nyt soi” -tietoa joko sisäänrakennetun rajapinnan tai lisäosan toteuttaman rajapinnan avulla. Taulukossa 2.1 on esitetty soittimet, jotka tarjoavat integroidun rajapinnan ”Nyt soi” -tiedon jakamiseen (Turney, s.a.). Taulukko sisältää kolme rajapintatyyppiä: MSN Messenger, Winamp ja iTunes COM. Tutustumme tässä luvussa ainoastaan MSN Messenger -rajapintaan. Kyseisen rajapinnan avulla on mahdollista saada suurimalta osalta soittimista ”Nyt soi” -tietoa, kun otetaan avuksi kolmansien osapuolien tekemät ohjelmat (lisäosat).

Yksi tällainen kolmansien osapuolien tarjoama ohjelma on Last.fm Scrobbler⁴. Scrobbler-ohjelman pääasiallinen tarkoitus ei kuitenkaan ole MSN Messenger -rajapinnan tarjoaminen. Sen tarkoitus on kerätä kappalekuuntelut soittimilta ja lähettää ne Last.fm-palveluun. Mutta Last.fm Scrobblerin avulla on myös mahdollista

¹<http://explore.live.com/windows-live-messenger>

²<http://www.pidgin.im/>

³<http://code.google.com/p/pidgin-musictracker/>

⁴<http://www.last.fm/download>

ottaa käyttöön MSN Messenger -rajapinta niissä soittimissa, joista kappalekuunte-
luta kerätään. Scrobblerin tukemat soittimet ovat foobar2000, iTunes, Winamp ja
Windows Media Player (*Which media players does the Last.fm Scrobbler support?*,
s.a.). Edellä mainituista soittimista ainakin foobar2000:een ja Winampiin on ole-
massa oma lisäosa, joka toteuttaa ainoastaan MSN Messenger -rajapinnan (*MSN
Messenger Now Playing Plugin*, s.a.).

Taulukko 2.1: Windowsissa toimivien soittimien rajapinnat.

Rajapinta	Soitin
MSN Messenger	Spotify
	VLC
	Windows Media Player
	Media Player Classic
Winamp	Winamp
	MediaMonkey
	AIMP2
iTunes COM	iTunes

2.1.2 MSN Messenger -rajapinta

Windows-käyttöjärjestelmässä toimivassa pikaviestimessä Live Messenger on omi-
naisuus, jonka avulla tuttaville on mahdollista näyttää mitä kappaletta kuunte-
lee. Tätä ominaisuutta varten tehtyä rajapintaa hyödyntäen soittimet voivat lähet-
tää pikaviestimelle tietoa soivasta kappaleesta. Toteutukseltaan rajapinta pohjautuu
WM_COPYDATA viestinvälitysmekanismiin, jonka avulla Windowsissa ikkunaso-
vellukset voivat lähettää viestejä toisilleen (Büscher, 2004).

Sovellus, joka vastaanottaa ”Nyt soi” -tietoa, luo ikkunan nimeltä
MsnMsgrUIManager. Soitin, joka lähettää tietoa soivasta kappaleesta, etsii funktiol-
la *FindWindowsEx* ikkunat nimeltä *MsnMsgrUIManager*. Soitin lähettää löydetuille
sovelluksille kappaletiedot funktion *SendMessage* avulla.

Lisäosan Music Tracker lähdekoodin tiedostosta *msn-compat.c*⁵ saadaan selvil-
le muodot, joissa soittimet kappaletietoa lähettävät. Kappaletieto voi olla neljässä
eri muodossa. Muodot ovat esitetty alla olevassa listassa. Toinen listan kohta on
Spotifyn käyttämä.

1. `\0Music\0enabled\0artist\0title\0`
2. `\0Music\0enabled\0format\0title\0artist\0`
3. `\0Music\0enabled\0format\0title\0artist\0album\0`

⁵<http://code.google.com/p/pidgin-musictracker/source/browse/trunk/src/msn-compat.c?r=468>

4. \0Music\0enabled\0format\0title\0artist\0album\0uuid\0

Arvo *enabled* sisältää numeron yksi, kun soittimella kuunnellaan kappaletta ja nollan, kun toisto on pysäytetty. Arvo *format* sisältää tekstin "{0} - {1}", jolla ilmaistaan kappaleen (*title*) ja artistin (*artist*) sisältö. Sen avulla on mahdollista kääntää teksti *title* - *artist* toisin päin, jos *title* esimerkiksi sisältäisi artistin nimen ja päin vastoin. Arvo *title* sisältää kappaleen nimen, *artist* artistin nimen ja *album* albumin nimen. Arvo *uuid* (Universally Unique IDentifier) sisältää universaalisti yksilöllisen tunnisteen.

MSN Messenger -rajapintaan saattaa liittyä eräs ongelma, johon tiedon vastaanottaja ei pysty mitenkään vaikuttamaan. Ongelma liittyy siihen, miten lähettäjä etsii vastaanottajat funktiolla *FindWindowsEx*. Jos lähettäjä etsii vain ensimmäisen ikkunan nimeltä *MsnMsgrUIManager*, tällöin muut samannimiset ikkunat eivät saa WM_COPYDATA -viestejä. Oikeaoppinen tapa etsiä ikkunat on hakea *while*-silmukassa kaikki ikkunat ja lähettää viestit kaikille. (Turney, s.a.)

2.1.3 Linux

Linux-käyttöjärjestelmissä suureen osaan musiikkisoittimista on toteutettu D-Bus -rajapinta (Desktop Bus), jonka avulla ulkopuoliset sovellukset pystyvät vaikuttamaan soittimen toimintaan. Sen avulla on esimerkiksi mahdollista hakea soittimelta kuunneltavan kappaleen tiedot.

Merkittävimmät työpöytäympäristöt, kuten Gnome⁶ ja KDE 4⁷, käyttävät D-Busia prosessien väliseen kommunikointiin (*Top Ten Distributions*, s.a.). Tästä johtuen myös suosituimmat musiikkisoittimet, jotka ovat tehty joko Gnomelle tai KDE 4:lle, tukevat pääsääntöisesti D-Busia. Aiemmissa KDE:n versioissa on ollut käytössä DCOP (Desktop COmmunication Protocol), joka on neljänneistä versiosta lähtien korvattu D-Busilla (Schoenmakers, 2009).

Taulukossa 2.2 on esitettynä joukko yleisimpiä Linuxissa toimivia musiikkisoittimia jaoteltuna toteuttamiensa rajapintojen mukaan. Suurin osa soittimista tarjoaa D-Busiin pohjautuvan MPRIS-määrittelyn mukaisen rajapinnan. Osa kuitenkin toteuttaa D-Bus -rajapinnan itse kehitetyn määrittelyn mukaisesti, kuten Ubuntuissa oletuksena käytettävä musiikkisoitin Banshee (Mathews, 2011). Amarokin versio 1.4 toteuttaa ainoana DCOP-rajapinnan. (Turney, s.a.)

On olemassa myös soittimia, joihin on lisäosan avulla saatavissa MPRIS-määrittelyn toteuttava rajapinta. Tällaisia soittimia ovat muun muassa Exaile⁸, MPD⁹, Song-

⁶<http://www.gnome.org/>

⁷<http://www.kde.org/>

⁸<http://www.exaile.org/>

⁹http://mpd.wikia.com/wiki/Music_Player_Daemon_Wiki

bird¹⁰ ja XMMS2¹¹. Soittimessa Exaile lisäosa tulee mukana. Se täytyy kuitenkin laittaa päälle asetuksista, jos haluaa tuen MPRIS-määrittelykselle.

Taulukko 2.2: Linuxissa toimivien soittimien tukemat rajapinnat.

Rajapinta	Soittimet
MPRIS (D-Bus)	Amarok 2
	Audacious
	Clementine
	Decibel
	Rhythmbox
	VLC
D-Bus	Banshee
	Listen
	Quod Libet
DCOP	Amarok 1.4

2.1.4 MPRIS-rajapinta

D-Bus-viestinvälitysmekanismeja hyödyntäen Linux-työpöytäympäristöihin on toteutettu määrittely nimeltä MPRIS¹², joka tarjoaa standardin rajapinnan mediasoitinien toiminnan hallintaan. Määrittelystä on olemassa kaksi versiota 1.0 ja 2.1. Keskitymme tässä määrittelyn versioon 1.0, koska se on toteutettu lähestulkoon kaikkiin uusimpiin soittimiin Linuxissa.

MPRIS-määrittelyn mukaan jokaisen soittimen tulee rekisteröidä itsensä D-Bus-palvelimeen nimellä, jonka alkuosa on aina *org.mpris*. Tämän yhteisen alkuosan perään liitetään soittimen nimi, jolloin nimi on *org.mpris.clementine*, kun käytetään soitinta Clementine¹³. Taulukossa 2.3 on esitetty rajapinnan tarjoamat objektit, kun soitin toteuttaa kokonaisuudessaan version 1.0 mukaisen MPRIS-määrittelyn. (*MPRIS D-Bus Interface Specification*, 2008)

Taulukko 2.3: Rajapinnan *org.freedesktop.MediaPlayer* tarjoamat objektit.

Objektin polku	Kuvaus
/	Juuri-objekti tarjoaa tietoa soittimen ja MPRIS-määrittelyn versiosta.
/Player	Tarjoaa metodeja soiton hallintaan ja soivan kappaleen metatietojen hakemiseen, ja signaaleja soiton tilan muutoksista.
/TrackList	Tarjoaa metodeja soittimen kappalelistan hallintaa.

¹⁰<http://getsongbird.com/>

¹¹http://xmms2.org/wiki/Main_Page

¹²<http://www.mpris.org/>

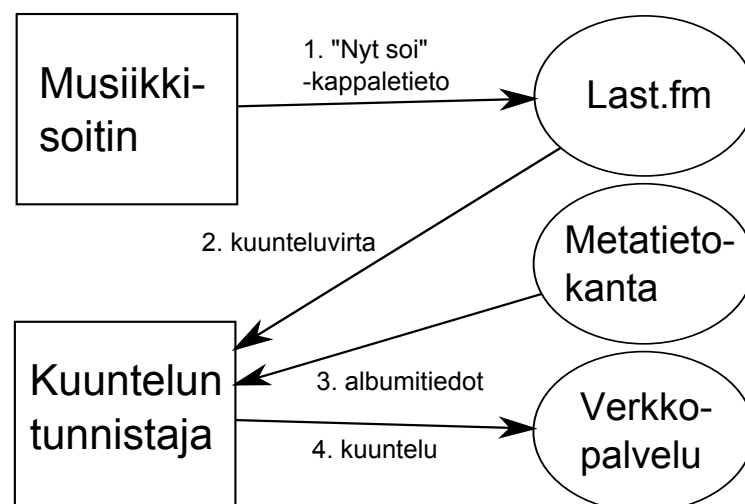
¹³<http://www.clementine-player.org/>

MPRIS-määrittystä käytetään rajapinnan *org.freedesktop.MediaPlayer* kautta. Rajapinnan objektin */Player* avulla on mahdollista tarkkailla kappaleen vaihtumista ja hakea soivan kappaleen tiedot uuden kappaleen yhteydessä. Objekti tarjoaa signaalin *TrackChange*, joka lähetetään, kun uusi kappale alkaa soimaan. Kappaleen tiedot saadaan haettua metodilla *GetMetadata*.

D-Busia voidaan käyttää Linuxissa suoraan komentoriviltä esimerkiksi ohjelman *qdbus* avulla. Kuunneltavan kappaleen tietojen hakeminen *qdbus*-ohjelman avulla soittimelta Clementine tapahtuu seuraavasti: ensimmäiseksi komentoriviparametriksi asetetaan soittimen bus-nimi, joka on *org.mpris.clementine*, toinen parametri on rajapinnan objektin nimi */Player* ja kolmas on rajapinnan metodi *org.freedesktop.MediaPlayer.GetMetadata*. (Sansome, s.a.). Metodin *GetMetadata* palauttaman kappaleen sisältökuvauksen tärkeimpiä attribuutteja ovat *artist*, *album*, *title*, *tracknumber*, *year*, *date* ja *genre*. Muut attribuutit eivät ole aivan niin oleellisia kuuntelun tunnistajan tai Statsterin kannalta (*MPRIS v1.0 Metadata guidelines*, s.a.).

2.1.5 Last.fm

Statsteriin on jo mahdollista lisätä albumikuunteluita Last.fm-kuunteluvirtaa hyödyntäen (kohta 5.1). Tätä samaista virtaa hyödyntäen kuunteluntunnistajan on mahdollista saada ”Nyt soi” -tietoa epäsuorasti käyttäjän soittimelta Last.fm:n palvelurajapintaa hyödyntäen. Kuvassa 2.2 on esitetty albumikuuntelun tunnistamisen vaiheet Last.fm-kuunteluvirran avulla. Kuvan suorakulmiot ilmaisevat paikallisia, käyttäjän koneella sijaitsevia sovelluksia ja soikiot ulkopuolisia, Internetissä toimivia verkkopalveluita.



Kuva 2.2: Kuuntelun tunnistaminen Last.fm-palvelun kuunteluvirtaa hyödyntäen.

Kappaletiedot Last.fm:ään lähettävä ohjelma on saatavilla lähestulkoon kaikkiin musiikkisoittimiin käyttöjärjestelmästä riippumatta joko integroituna tai lisäosana (*Audioscrobbler Plugins*, s.a.). Jotta kuunteluntunnistaja pystyy hyödyntämään käyttäjän kuunteluvirtaa albumikuuntelun tunnistamisessa, täytyy sen tietää käyttäjän Last.fm-käyttäjänimi. Käyttäjänimi on ainut tarvittava tieto, olettaen että käyttäjän soitin lähettää tiedot kuunnelluista kappaleista Last.fm:ään.

Last.fm-palvelu tarjoaa rajapinnan, jonka avulla käyttäjän kuunteluvirta on mahdollista hakea. Kuunteluvirta sisältää tiedot kuunnelluista kappaleista. Kappaletiedot koostuvat muun muassa kappaleen nimestä, artistista ja albumista.

Verkkopalvelua Last.fm tullaan käsittelemään tarkemmin luvussa 3. Luvun kohdassa 3.1 näytetään, miten kuunneltavan kappaleen tiedot haetaan Last.fm-kuunteluvirrasta, ja miten sen jälkeen albumin kappaleista saadaan haettua kyseisestä palvelusta.

2.2 Albumin tunnistaminen

Kuunneltavan kappaleen tietojen saaminen on ensimmäinen askel kohti soitinriippumattonta albumikuuntelun tunnistusta. Toinen askel on tarvittavien tietojen hakeminen, jotta kuunneltava albumi voidaan tunnistaa. Tiedot haetaan metatietokannasta käyttäen apuna kappaleen nimeä ja artistia. Soittimelta saatujen tietojen lisäksi albumikuuntelun tunnistukseen tarvitaan albumin nimi ja kappaleista. Julkaisuvuosi on myös hyödyllinen Statsterin kannalta, koska sen avulla on mahdollista yksilöidä artistin samannimiset, mutta eri vuonna julkaistut albumit. Albumin kappaleistan avulla on mahdollista seurata kuuntelun etenemistä ja päätellä, koska albumi on kuunneltu kokonaan viimeiseen kappaleeseen asti.

Ensimmäisen kappaleen yhteydessä haettujen albumitietojen perusteella ei voida vielä olla varmoja miltä albumilta kuunneltava kappale on, koska kappale voi esiintyä monella eri albumilla. Kappale voi löytyä esimerkiksi studio-, kokoelma- ja live-albumilla. Tästä syystä albumin tunnistamisen apuna täytyy käyttää jo soitettujen kappaleiden yhteydessä saatuja albumitietoja. Aloitettaessa kuuntelemaan albumin ensimmäistä kappaletta kuunteluntunnistaja tietää vain kappaleen nimen ja artistin. Se ei tiedä miltä albumilta kappale on, eikä kappalenumeroa. Kuuntelun edetessä seuraavaan kappaleeseen kuunteluntunnistaja voi käyttää edellistä kappaletta apuna kuunneltavan albumin päättelemisessä. Toisen kappaleen yhteydessä haettujen albumien kappaleistoista voidaan tarkistaa millä albumilla kuunnellut kappaleet ovat peräkkäin. Albumi, jolla kappaleet ovat peräkkäin, on todennäköisimmin albumi, jota kuunnellaan. Kuuntelun edetessä kappalehistoriaa hyödyntäen voidaan olla yhä varmempia albumin oikeellisuudesta, jos kappaleet esiintyvät peräkkäisessä järjestyksessä albumilla.

2.2.1 Esimerkki: Nirvanan albumin Nevermind kuunteleminen

Tässä alakohdassa tutustutaan albumin tunnistuksen teoriaan käytännönläheisemmin käyttämällä apuna artistin *Nirvana* studioalbumin *Nevermind* kuuntelemista. Albumin ensimmäinen kappale on *Smells Like Teen Spirit*. Kuvitteellinen metatietokanta, josta kappaleille haetaan oikeata albumia, sisältää myös muita Nirvanan albumeita, kuten livealbumit *From the Muddy Banks of the Wishkah* ja *MTV Unplugged in New York* ja kokoelma-albumin *Nirvana*. Kyseiset albumit sisältävät samoja kappaleita kuin albumi *Nevermind*.

Ensimmäisen kappaleen yhteydessä metatietokannasta etsitään kaikki artistin *Nirvana* albumit, jotka sisältävät kappaleen *Smells Like Teen Spirit*. Tuloksena saadaan kolme albumia, joilla kappale esiintyy, joten vielä ei voida olla varmoja oikeasta albumista (albumi *MTV Unplugged in New York* ei sisällä kyseistä kappaletta). Kuuntelun edetessä seuraavaan *In Bloom* -kappaleeseen, metatietokannasta saadaan kaksi albumia *Nirvana* ja *Nevermind*. Nyt kun käytetään kappalehistoriaa apuna albumin tunnistamisessa, huomataan, että kokoelma-albumilla *Nirvana* kappale *Smells Like Teen Spirit* on viidentenä ja nykyinen kappale *In Bloom* kahdeksantena. Tästä syystä kokoelma-albumi *Nirvana* ei ainakaan ole kuunneltava albumi. Tämän jälkeen käydään läpi albumin *Nevermind* kappaleista, huomataan, että kappale *Smells Like Teen Spirit* on ensimmäisenä ja *In Bloom* toisena. Tehtyjen vertailujen ansiosta voidaan jo olla melko varmoja siitä, että kuunneltava albumi on *Nevermind*.

Kolmannen kappaleen *Come as You Are* alkaessa soimaan voidaan joko käyttää *Nevermind* albumia vertailualbumina tai tarkistaa vielä mitä albumeita sillä löytyy. Kappaleella löytyy kolme albumia: *MTV Unplugged in New York*, *Nevermind* ja *Nirvana*. Studioalbumi *Nevermind* on taas ainut, jolta kuunnellut kappaleet löytyvät järjestyksessä.

Seuraavan kappaleen *Lithium* yhteydessä voidaan jo käyttää vertailujulkaisuna *Nevermind*-albumia. Albumi voidaan todeta kuunnelluksi ja lähettää Statsteriin, kun kuuntelu on edennyt vertailujulkaisun kappalelistan mukaisessa järjestyksessä viimeiseen kappaleeseen saakka.

2.2.2 Albumin tunnistamisessa käytettävät metatietokannat

Kuunneltavan albumin tunnistamisessa käytettävät metatietokannat ovat jaettu paikallisiin ja ulkoisiin metatietokantoihin. Paikallinen metatietokanta sijaitsee käyttäjän koneella. Paikallinen metatietokanta saadaan luotua lukemalla läpi käyttäjän koneella olevien musiikkitiedostojen sisältökuvaukset. Yksi esimerkki tällaisesta paikallisesta metatietokannasta on soitinohjelmiston ylläpitämä musiikkikirjasto, jonka avulla Statsterin nykyiset lisäosat toimivat (kohta 5.3). Ulkoisille metatietokannoille

ominaista on se, että ne ovat jonkin ulkoisen tahon luomia yleensä Internetissä toimivia verkkopalveluita. Tällaisia musiikkia varten luotuja metatietokantapalveluita ovat muun muassa Last.fm, MusicBrainz ja Discogs¹⁴.

Kumpaankin metatietokantatyyppiin liittyy sekä etuja että haittoja. Yksi ulkoisen metatietokannan etu verrattuna paikalliseen on sen käyttöönoton helppous. Ulkoinen metatietokanta on valmiiksi olemassa, joten toteutettavaksi jää vain kyselyiden tekeminen metatietokantaan. Paikallinen metatietokanta täytyy luoda osaksi kuunteluntunnistajaa lukemalla läpi käyttäjän musiikkitiedostojen sisältökuvaukset. Toinen etu ulkoisessa on se, että käyttäjän ei tarvitse kuunnella musiikkia omalta koneelta. Musiikki voi tulla Internetistä virtana (stream), esimerkiksi palvelusta Spotify.

Paikallisen metatietokannan etuja verrattuna ulkopuoliseen on tiedon yhdenmukaisuus ja luotettavuus. Soittimelta saatavat kappaleetiedot vastaavat täysin paikallisessa metatietokannassa olevia tietoja, koska metatietokanta on luotu kuunneltavien kappaleiden sisältökuvauksista. Paikallinen metatietokanta on luotettava siinä mielessä, että se ei ole riippuvainen verkkoyhteydestä. Se on aina käytettävissä.

Mahdollisimman luotettavan tunnistamisen ja mahdollisimman usean soittimen käytön kannalta ulkoisen ja paikallisen metatietokannan yhdistelmä olisi paras vaihtoehto. Paikalliset metatietokannat ovat kuitenkin rajattu tämän diplomityön ulkopuolelle. Tässä työssä tullaan tarkastelemaan tarkemmin vain ulkoisia metatietokantoja.

2.3 Tiedon haun teknologiat Internetin metatietopalveluissa

Verkkopalvelut, jotka tarjoavat musiikkiin keskittyneen metatietokannan, käyttävät pääsääntöisesti samoja teknologioita palvelurajapinnoissaan. Palvelurajapinta mahdollistaa ohjelmallisen tiedonhaun.

Tässä kohdassa tullaan käsittelemään lyhyesti tärkeimmät teknologiat, joiden avulla tiedon hakeminen onnistuu Internetin metatietokannoista. Ensin tutustutaan XML-kuvauskieleen, joka mahdollistaa ohjelmallisen tiedonkäsittelyn. XML-tietoa puolestaan siirretään verkkopalvelusta HTTP-protokollan avulla sovellukselle. Lopuksi tutustutaan kahteen arkkitehtuurityyliin, joiden avulla palvelurajapintojen käyttö on selkeämpää ja yhdenmukaisempaa.

2.3.1 XML

XML on kuvauskieli, jonka avulla tieto esitetään rakenteisessa muodossa. XML mahdollistaa ohjelmallisen tiedonkäsittelyn. Sitä käytetään yleisesti sovellusten välisessä tiedon siirrossa. XML-muotoista tietoa on suhteellisen helppo lukea ohjelmallisesti,

¹⁴<http://www.discogs.com/>

koska XML-ohjelmakirjastot ovat yleisiä.

Tässä työssä käsiteltävät Internetissä toimivat metatietokanta-palvelut välittävät tietoa XML-kuvauskieltä hyödyntäen. Kun palvelun käyttämä XML-formaatti on selvillä, saadusta tiedosta on helppo lukea tarvittavat tiedot.

XML-dokumentin formaatti on vapaasti muodostettavissa, tiettyjen perussääntöjen puitteissa. XML-muotoinen dokumentti alkaa aina prologilla: `<?xml version="1.0" encoding="utf-8"?>`. XML-muotoinen dokumentti muodostuu elementeistä, joiden nimet ovat kulmasulkeiden sisällä: `<elementin-nimi/>`. Tyhjän elementin nimi loppuu aina vinoviivaan. Elementti voi sisältää attribuutteja: `<elementi-nimi attribuutin-nimi="arvo"/>`, attribuutteja voi olla elementillä rajoittamaton määrä. XML-elementit voivat koostua muista XML-elementeistä: `<elementin-nimi><lapsi-elementti/></elementin-nimi>`. Tällöin sisällön rajaa-va lopetus-elementti alkaa aina vinoviivalla ja sisältää vain nimen. Elementti voi myös sisältää tekstimuotoista tietoa, jota elementin nimellä ja attribuuteilla kuvataan. Elementin sisällä olevia elementtejä kutsutaan lapsi-elementeiksi, ne voivat puolestaan sisältää omia lapsia jne. (*Extensible Markup Language (XML) 1.0 (Fifth Edition)*, 2008)

2.3.2 HTTP

HTTP on Internetin yksi tärkeimmistä tiedonvälitysprotokollista. Se toimii protokollapinon sovelluskerroksella kuljetuskerroksen päällä (*OSI model*, s.a.). Kuljetuskerroksen protokolla on tässä työssä aina TCP. HTTP:n avulla siirretään esimerkiksi XML-muotoista tietoa sovellusten ja palvelimien välillä.

HTTP:n toiminta perustuu pyyntöihin (request) ja vastauksiin (response). Pyynnöillä haetaan tietoa palvelimelta tai tallennetaan sitä, palvelin vastaa saamaansa pyyntöön HTTP-vastauksella. Vastaus sisältää pyydettyt tiedot, tai virheen sattuessaa protokolla ilmoittaa tapahtuneesta virheestä. HTTP-pyynnöt ovat yleensä joko GET- tai POST-tyyppisiä. Metodien käyttötarkoituksiin perehdytään tarkemmin seuraavassa alakohdassa REST.

HTTP-protokollan rakenne koostuu otsikko-osasta (header) ja runko-osasta (body). Pyynnön otsikko kertoo muun muassa käytettävän metodin. Vastauksen otsikko sisältää puolestaan tietoa pyynnön onnistumisesta, palvelimesta ja sisällöstä. Runko-osa on tiedon kuljetusta varten asiakkaan ja palvelimen välillä. Rungossa siirretään esimerkiksi POST-metodilla palvelimelle tieto ja palvelin palauttaa saamansa pyynnön perusteella tietoa HTTP-vastauksen rungossa. (*Hypertext Transfer Protocol – HTTP/1.1*, 1999) Yleensä palvelimelta saatu tieto on XML-muotoista, kuten tässä työssä.

2.3.3 REST

REST on ohjelmistoarkkitehtuurityyli, jota hyödynnetään yleisesti HTTP-protokollaa käytettäessä. Se luo säännöt, miten esimerkiksi GET- ja POST-metodeja tulee käyttää asiakaspalvelin-mallin mukaisessa tiedon välityksessä. Kyseisten metodien lisäksi harvinaisemmille PUT- ja DELETE-metodeille REST määrittää myös omat käyttötarkoitukset. (Fielding, 2000)

HTTP:n metodi GET on tarkoitettu tiedon hakuun palvelimelta. Sen avulla ei muuteta palvelun tilaa. Pyyntö GET koostuu palvelimen URL-osoitteesta, jolla yksilöidään haluttu resurssi. Osoitteeseen voidaan myös liittää GET-parametreja, joilla voidaan tarkentaa resurssin palauttamaa sisältöä.

POST-metodia on REST-periaatteen mukaan käytettävä vain tiedon tallentamiseen palvelimelle. POST-metodin avulla palvelun tilaa on mahdollista muuttaa. POST-pyyntö koostuu resurssin URL-osoitteesta. Osoitteeseen ei liitetä lisäparametreja. Metodin POST parametrit tulevat HTTP:n runko-osaan.

Seuraavassa luvussa tutustutaan verkkopalveluihin Last.fm ja MusicBrainz, jotka tarjoavat REST-periaatteen mukaisen palvelurajapinnan. Tässä työssä käytetään ainoastaan GET-metodia, koska edellä mainituista palveluista haetaan vain tietoa. Ennen lukua 3 perehdytään kuitenkin vielä Last.fm-palvelun tarjoamaan vaihtoehtoiseen arkkitehtuurityyliin nimeltä XML-RPC.

2.3.4 XML-RPC

XML-RPC on etäkutsuprotokolla (remote procedure call), joka käyttää XML-kuvauskieltä tiedon esittämiseen ja HTTP-protokollaa tiedon siirtämiseen. Protokollan POST-metodia käytetään tiedon välityksessä (Winer, 1999).

Siirrettävä tieto kuvataan XML:ää apuna käyttäen, jotta ohjelmallinen tiedon käsittely on mahdollista. XML-tieto siirretään HTTP:n runko-osassa, koska käytettävä metodi on POST. XML-RPC ei noudata tämän vuoksi REST periaatetta.

3. AVOIMET MUSIIKIN METATIETOKANNAT

Avoimet musiikin metatietokannat ovat ulkoisia Internetissä toimivia palveluita, joita voi vapaasti kuka tahansa käyttää. Ne tarjoavat tietoa esimerkiksi artistien albumeista ja kappaleista. Avoimuudella ei tässä tarkoiteta metatietokannan sisällön vapaata muokattavuutta eli sisältöön ei välttämättä pysty itse vaikuttamaan.

Avoimia metatietokantoja on Internetissä ainakin neljä: Last.fm, MusicBrainz, Discogs ja Freedb¹, joista kahta ensimmäistä käsitellään tarkemmin tässä luvussa. Luetelluista palveluista kahta viimeistä ei käsitellä, koska niiden palvelurajapinnat eivät ole riittävän monipuoliset. Discogs ei tarjoa suoraa kappalehakua, jolla saataisiin kappaleen julkaisu (*Discogs API Documentation*, s.a.). Freedb:n rajapinta puolestaan ei tarjoa suoraa kappalehakua. Rajapinnan kautta voidaan hakea tietoa vain CD-levystä laskettavan yksilöllisen tunnisteiden avulla (*What is freedb?*, s.a.).

Valittuja palveluita tullaan tarkastelemaan albumikuuntelun tunnistamisen näkökulmasta eli miten palvelurajapintoja käyttäen kappaleelle saadaan haettua albumi ja sen kappaleista, kun tiedossa on vain kappaleen nimi ja artisti. Ensimmäisenä käsitellään palvelua Last.fm ja sen jälkeen MusicBrainzia.

3.1 Last.fm

Last.fm on alkujaan verkkopalvelu, joka on tehty helpottamaan uuden musiikin löytämistä ja omien kuuntelutottumuksien seuraamista. Tästä se on kehittynyt laajaksi musiikin metatietokannaksi.

Palvelu pohjautuu käyttäjien luomaan sisältöön. Kun palvelun käyttäjät kuuntelevat musiikkia soittimillaan, joissa on Last.fm Scrobbler -asiakasohjelma, he samalla lähettävät kuunnelluista kappaleista metatietoa palveluun. Last.fm Scrobbler -asiakasohjelma on palvelun virallisesti tarjoama sovellus. Virallisen lisäksi on olemassa epävirallisia Scrobbler-ohjelmia sekä integroituna suoraan soittimeen että li säosien avulla käyttöönotettavissa (*Scrobblers*, s.a.).

Suuri kuuntelunlähettimien määrä on seurausta hyvästä palvelurajapinnasta, joka mahdollistaa palveluun liittymisen ohjelmallisesti asiakasohjelmista, ja myös suuri asiakasohjelmien määrä kertoo palvelun suuresta suosiosta. (*Last.fm statistics*, 2008) Kappalekuuntelujen vastaanoton lisäksi Last.fm:n palvelurajapinta tarjoaa kattavan rajapinnan metatiedon hakemiseen. (*Last.fm Web Services*, s.a.)

¹<http://www.freedb.org/>

3.1.1 Palvelurajapinta

Last.fm-palvelu tarjoaa monipuolisen rajapinnan, jonka avulla palvelusta saadaan tietoa ulos XML-muodossa. Tietoa on mahdollista hakea joko REST-periaatteen mukaisesti tai XML-RPC-pyynnöillä. Tutustumme tässä luvussa vain REST-periaatteen mukaiseen tiedon hakuun.

Palvelurajapinnan kautta haettava tieto on jaettu luokkiin sisällön perusteella, ja jokainen luokka puolestaan metodeihin, joilla hakua yksilöidään tarkemmin. Palvelurajapinnan tarjoamat tietokokonaisuudet eli luokat ovat: *album*, *artist*, *auth*, *chart*, *event*, *geo*, *group*, *library*, *radio*, *tag*, *tasteometer*, *track*, *user* ja *venue*. Albumikuuntelun tunnistamisen kannalta tärkeimmät luokat ovat *album* ja *track*, ja myös *user* jos käyttäjän kuuntelemien kappaleiden tiedot haetaan suoraan Last.fm-palvelusta. Taulukossa 3.1 on nähtävillä yksi luokan *user* metodeista. Metodilla *getRecentTracks* saadaan haettua kuunneltujen kappaleiden tiedot, edellyttäen että käyttäjän soitin lähettää kappalekuuntelut palveluun.

Taulukko 3.1: Luokan *user* metodi, jolla saadaan kuunneltavan kappaleen tiedot.

Metodi	Kuvaus
<i>getRecentTracks</i>	Palauttaa listan käyttäjän viimeksi kuuntelemista kappaleista. Lista sisältää myös kuunneltavan kappaleen, tällöin siihen on liitetty attribuutti <i>nowplaying</i> , jonka arvo on true.

Taulukossa 3.2 on esitettyä luokan *track* hakumetodit. Luokka sisältää myös muita metodeja, mutta ne eivät ole oleellisia albumikuuntelun tunnistamisessa, joten ne on jätetty esittämättä.

Taulukko 3.2: Luokan *track* hakumetodeja.

Metodi	Kuvaus
<i>getInfo</i>	Palauttaa kappaleeseen liittyvän metadatan. Pakolliset parametrit ovat <i>artist</i> jos ei <i>mbid</i> ole annettu, <i>track</i> jos ei <i>mbid</i> ole annettu, ja <i>api_key</i> . Vaihtoehtoisia parametreja ovat <i>autocorrect</i> , antamalla sen arvoksi 1 saadaan korjatut nimet artistille ja kappaleelle, <i>mbid</i> ja
<i>search</i>	Palauttaa hakua vastaavat kappaleet relevanttiuden mukaisesti järjestettynä. Pakolliset parametrit ovat <i>track</i> ja <i>api_key</i> , vaihtoehtoisia ovat <i>limit</i> , <i>page</i> ja <i>artist</i> .

Albumikuuntelun tunnistamisen kannalta tarpeellisimmat *track*-luokan metodit ovat *getInfo* ja *search*. Metodin *getInfo* avulla saadaan haettua kappaleen albumit käyttäen parametreja *artist* ja *track*. Metodilla *search* voidaan etsiä kappaleita palvelusta jos ei olla varmoja kappaleen kirjoitusasun oikeellisuudesta. Tämä tehdään,

koska metodi *getInfo* ei välttämättä palauta kappaleelle albumia jos kappaleen kirjoitusasu on väärä.

Taulukossa 3.3 on esitetty luokan *album* hakumetodit. Luokka sisältää myös muita metodeja, mutta ne eivät ole oleellisia työn kannalta. Luokan *album* metodologia *getInfo* käytetään albumin kappaleiden hakemiseen. Metodia varten tiedossa täytyy olla albumin oikea nimi ja albumin artisti.

Taulukko 3.3: Luokan *album* hakumetodeja.

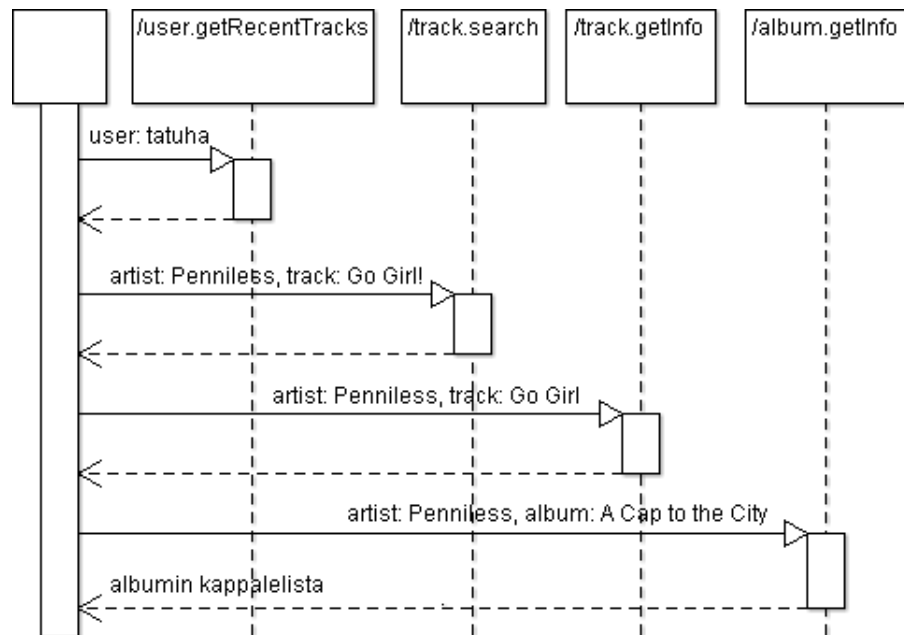
Metodi	Kuvaus
<i>getInfo</i>	Palauttaa albumiin liittyvän metadatan. Pakolliset parametrit ovat <i>artist</i> jos ei <i>mbid</i> ole annettu, <i>album</i> jos ei <i>mbid</i> ole annettu, ja <i>api_key</i> . Vaihtoehtoisia parametreja ovat <i>autocorrect</i> , jos sille on annettu arvoksi 1 saadaan korjattu nimi artistille, <i>mbid</i> , <i>username</i> ja <i>lang</i> .
<i>search</i>	Palauttaa kaikki albumit, jotka vastaavat hakua. Tulokset on järjestetty relevanttiuden mukaan. Pakolliset parametrit ovat <i>album</i> ja <i>api_key</i> , ja vaihtoehtoisia <i>limit</i> ja <i>page</i> .

Palvelurajapinnan tarjoamia metodeja käytetään osoitteen <http://ws.audioscrobbler.com/2.0/> kautta. Haettaessa palvelusta tietoa REST-periaatteen mukaisesti liitetään luokka ja luokan metodi osoitteeseen GET-parametrin *method* arvoksi. Luokka ja metodi erotetaan pisteellä toisistaan seuraavasti: `method=luokka.metodi`. Luokan tarjoamien metodien parametrit liitetään GET-metodin parametreiksi seuraavasti: `method=luokka.metodi&artist=artisti&track=kappale`.

3.1.2 Esimerkki: albumin kappalelistan hakeminen

Tässä esimerkissä tutustutaan siihen, miten Last.fm-palvelua hyödyntäen saadaan kuunneltavan kappaleen albumi ja kappalelista, kun soitin lähettää tiedot kuunnelluista kappaleista palveluun. Albumin tiedot haetaan kappaleen nimen ja artistin avulla.

Kuunneltava kappale on *Go Girl!* artistin *Penniless* albumilta *A Cab To the City*. Albumi on sen verran uusi – julkaistu 31 elokuuta 2010 – ettei sitä löydy esimerkiksi MusicBrainzista (tarkistettu 18.7.2011). Kuvassa 3.1 on nähtävillä kaikki tarvittavat metodikutsut, jotka Last.fm-palvelurajapinnalle on tehtävä albumin tietojen saamiseksi. Kutsujen sarja alkaa kuunneltavan kappaleen nimen ja artistin hakemisella ja päättyy halutun albumin kappalelistan saamiseen.



Kuva 3.1: Last.fm-palvelurajapinnan metodikutsut ja niiden suoritusjärjestys.

Käyttäjän kuunteleman kappaleen tiedot saadaan palvelurajapinnan tarjoamalla metodilla *user.getRecentTracks*. Metodi tarvitsee parametreikseen vain Last.fm-käyttäjänimen, tässä tapauksessa *tatuha*, ja rajapinta-avaimen (API key). URL-osoite on tällöin seuraavanlainen:

```
http://ws.audioscrobbler.com/2.0/?method=user.getRecentTracks&api_key=e00a410eee50a17eac82f397d6b5b41b&user=tatuha
```

Listauksessa 3.1 on esitetty osa yllä olevan pyynnön palauttamasta vastauksesta. Elementin *recenttracks* attribuutista *perPage* (rivi 2) nähdään kuinka monta kappaletta eli *track*-elementtiä vastaus todellisuudessa sisältää. Vastauksesta on näkyvillä vain ensimmäinen kappale. Kyseistä kappaletta kuunnellaan juuri nyt, koska elementin *track* attribuuttina on *nowplaying* ja sen arvo on *true* (rivi 3). Elementin *track* sisällä *artist*-elementissä on kuunneltavan kappaleen artisti (rivi 5), ja elementin *name* sisällä on kappaleen nimi (rivi 7). Kappaleelle ei ole saatavilla albumia, koska kuunneltavan kappaleen kirjoitusasu ei ole Last.fm:n mukaan oikea (rivi 10).

Listaus 3.1: Metodin *track.getRecentTracks* palauttama kappaleetieto.

```

1 <lfm status="ok">
2   <recenttracks user="tatuha" page="1" perPage="10" totalPages="
    11189" total="11188">
3     <track nowplaying="true">
4       <artist mbid="583f329b-9b42-4a54-b957-067d2d79ddc8">
5         Penniless
6       </artist>
7       <name>Go Girl!</name>

```



```

8      <streamable>1</streamable>
9      <mbid/>
10     <album mbid=""/>
11     <url>http://www.last.fm/music/Penniless/_/Go+Girl%21</url>
12   </track>
13 </recenttracks>
14 </lfm>

```

Kun luokan *user* metodi *getRecentTracks* ei palauta kuunneltavan kappaleen albumia, voidaan kappaleen oikea kirjoitusasu selvittää luokan *track* metodilla *search*, jota voidaan kutsua seuraavasti:

```
http://ws.audioscrobbler.com/2.0/?method=track.search&api_key=
e00a410eee50a17eac82f397d6b5b41b&artist=Penniless&track=Go+Girl!
```

Osoitteelle annetaan metodin ja rajapinta-avaimen lisäksi GET-parametrit *artist* ja *track*, joiden arvot saatiin listauksesta 3.1. Palvelurajapinta vastaa pyyntöön palauttamalla tiedot kappaleista, jotka vastaavat annettuja parametreja. Listauksessa 3.2 on esitettyä oleellisin sisältö haun palauttamasta vastauksesta. Hakutulokset ovat elementin *trackmatches* sisällä lapsi-elementeissä *track*. Tulokset ovat järjestetty siten, että kappale, jota on kuunneltu eniten Last.fm:ssä on ensimmäisenä. Tämä nähdään elementeistä *listeners* (riveillä 7 ja 15).

Listaus 3.2: Metodin *track.search* hakutulokset.

```

1 <trackmatches>
2   <track>
3     <name>Go Girl</name>
4     <artist>Penniless</artist>
5     <url>http://www.last.fm/music/Penniless/_/Go+Girl</url>
6     <streamable fulltrack="0">1</streamable>
7     <listeners>113</listeners>
8     <mbid/>
9   </track>
10  <track>
11    <name>Go Girl!</name>
12    <artist>Penniless</artist>
13    <url>http://www.last.fm/music/Penniless/_/Go+Girl%21</url>
14    <streamable fulltrack="0">1</streamable>
15    <listeners>39</listeners>
16    <mbid/>
17  </track>
18 </trackmatches>

```

Palvelurajapinnan tarjoaman kappalehaun (*track.search*) avulla saatiin selvitettyä kuunneltavan kappaleen kirjoitusasu, joka on *Go Girl* ilman huutomerkkiä. Kirjoitusasun selvittäminen on tärkeää, sillä virheellisellä nimellä ei saada albumille

kappalelistaa. Kun kappaleen oikea nimi on tiedossa, voidaan selvittää albumi, jolle kappale kuuluu. Albumi saadaan selville luokan *track* metodilla *getInfo*. Palvelurajapinnalle lähetettävä pyyntö on seuraavanlainen.

```
http://ws.audioscrobbler.com/2.0/?method=track.getInfo&api_key=
e00a410eee50a17eac82f397d6b5b41b&artist=Penniless&track=Go+Girl
```

Palvelu vastaa saamaansa *track.getInfo*-pyyntöön palauttamalla kappaleen tiedot XML-muodossa. Listauksessa 3.3 on esitetty oleellisin osa saadusta vastauksesta. Elementin *album* lapsesta *title* nähdään kuunneltavan kappaleen albumin otsikko *A Cab to the City* (rivi 17).

Listaus 3.3: Metodin *track.getInfo* palauttavat tiedot.

```

1 <track>
2   <id>439035452</id>
3   <name>Go Girl</name>
4   <mbid/>
5   <url>http://www.last.fm/music/Penniless/_/Go+Girl</url>
6   <duration>249000</duration>
7   <streamable fulltrack="0">1</streamable>
8   <listeners>113</listeners>
9   <playcount>438</playcount>
10  <artist>
11    <name>Penniless</name>
12    <mbid>583f329b-9b42-4a54-b957-067d2d79ddc8</mbid>
13    <url>http://www.last.fm/music/Penniless</url>
14  </artist>
15  <album position="10">
16    <artist>Penniless</artist>
17    <title>A Cab to the City</title>
18    <mbid/>
19    <url>
20      http://www.last.fm/music/Penniless/A+Cab+To+the+City
21    </url>
22    <!-- Jätetty image-elementit pois tilaa viemästä -->
23  </album>
24 </track>
```

Kun kuunneltavan kappaleen albumi on tiedossa, voidaan hakea albumin kappaleista käyttämällä palvelurajapinnan luokan *album* metodologia *getInfo*. Palvelulle lähetettävän pyynnön URL-osoite on seuraavanlainen:

```
http://ws.audioscrobbler.com/2.0/?method=album.getinfo&api_key=
e00a410eee50a17eac82f397d6b5b41b&artist=Penniless&album=A+Cab+To+the+
City
```

Listauksessa 3.4 on nähtävillä osa Last.fm-palvelun palauttamasta vastauksesta. Esillä on vain ensimmäinen ja viimeinen kappale albumin kappalelistasta. Kappaleita 2–9 ei näytetä tilanpuutteen vuoksi. Elementti *tracks* koostuu lapsielementeistä *track*, joiden attribuutti *rank* kertoo kappaleen järjestysnumeron albumilla. Elementti *track* sisältää kappaleeseen liittyvää metatietoa, kuten nimen ja artistin.

Listaus 3.4: Näkyvillä kappaleiden 1 ja 10 tiedot metodin *album.getInfo* palauttamasta kappalelistasta.

```

1 <tracks>
2   <track rank="1">
3     <name>Don't You Know It's Monday?</name>
4     <duration>238</duration>
5     <mbid/>
6     <url>http://www.last.fm/music/Penniless/_/Don%27t+You+Know+It
      %27s+Monday%3F</url>
7     <streamable fulltrack="0">1</streamable>
8     <artist>
9       <name>Penniless</name>
10      <mbid>583f329b-9b42-4a54-b957-067d2d79ddc8</mbid>
11      <url>http://www.last.fm/music/Penniless</url>
12    </artist>
13  </track>
14  <!-- Kappaleet 2-9 jätetty pois -->
15  <track rank="10">
16    <name>Go Girl</name>
17    <duration>249</duration>
18    <mbid/>
19    <url>http://www.last.fm/music/Penniless/_/Go+Girl</url>
20    <streamable fulltrack="0">1</streamable>
21    <artist>
22      <name>Penniless</name>
23      <mbid>583f329b-9b42-4a54-b957-067d2d79ddc8</mbid>
24      <url>http://www.last.fm/music/Penniless</url>
25    </artist>
26  </track>
27 </tracks>

```

3.2 MusicBrainz

MusicBrainz on avoimen yhteisön ylläpitämä vapaa musiikin metatietokanta. Sen historia juontaa juurensa 1990-luvun lopulle, kun Gracenote osti CDDB:n (Compact Disc Database). Verkkopalvelu CDDB oli tätä ennen avoin CD-levyihin metatietoa tarjoava palvelu. Gracenote kuitenkin kiristi CDDB:n käyttöehtoja. Tämän vuoksi syntyi avoin musiikin metatietokanta nimeltä MusicBrainz, joka ei enää tyytynyt tarjoamaan pelkästään CD-levyihin liittyvää metatietoa. (*MusicBrainz History*, s.a.)

Palvelun tietosisällön rakenne koostuu muun muassa seuraavanlaisista resursseista: artisteista, julkaisuista, kappaleista ja levymerkeistä. Resurssit puolestaan sisältävät erilaisia attribuutteja, joilla sisältöä kuvataan tarkemmin. Palvelu sisälsi työn kirjoitushetkellä 624 680 artistia, 923 429 julkaisua, 10 868 801 kappaletta ja 51 026 levymerkkiä (*MusicBrainz database statistics*, 2011).

MusicBrainzin sisältö on kaikkien palveluun rekisteröityneiden käyttäjien muokattavissa. Palveluun rekisteröityminen on avoin kaikille Internetin käyttäjille. Käyttäjien palveluun tekemät muutokset eivät tule kuitenkaan voimaan välittömästi, koska palvelussa on niin kutsuttu äänestämiskäytäntö, jolla pyritään varmistamaan sisällön oikeellisuus. Sisällön oikeellisuus ja se, että jokainen voi luoda uutta sisältöä, on erittäin tärkeä ominaisuus metatietokantapalvelulle, jotta se sisältäisi mahdollisimman kattavasti julkaisuja niin vanhoja kuin uusiakin. Kunnollinen palvelurajapinta on myös tärkeä edellytys kunnolliselle metatietokantapalvelulle.

3.2.1 Palvelurajapinta

MusicBrainz tarjoaa ohjelmalliseen tiedonhakuun monipuolisen rajapinnan, jonka avulla metatietokannan tietoa voivat hyödyntää ulkopuoliset sovellukset. Palvelurajapintaa on mahdollista käyttää joko suoralla haulla tai indeksoidulla haulla. Suorassa haussa resurssin tiedot haetaan yksikäsitteisen tunnuksen avulla (*MBID*). Indeksoitu haku puolestaan käyttää apunaan hakupalvelinta, jonka hakumoottorina käytetään Lucenea². Hakupalvelimen avulla tietoa on mahdollista hakea hakusanoilla. Kumpaaakin hakua käytetään HTTP-yhteyden välityksellä. Palvelurajapinta on suunniteltu REST periaatteen mukaiseksi. Rajapinta palauttaa tulokset XML-muotoisena UTF-8-merkkikoodattuna tekstinä. Vaihtoehtoisesti MusicBrainzin palvelurajapinta tarjoaa JSON-muotoisen tiedon esitystavan. (*XML Web Service*, s.a.)

MusicBrainzin tietokannan tietomalli on toteutettu olioperustaisesti. Sen pääluokat ovat *artist*, *recording*, *release*, *release-group*, *work* ja *label*. Jokaisella luokalla on omat attribuuttinsa ja suhteet muihin luokkiin. Esimerkiksi luokalla *artist* voi olla monta julkaisua (*release*), ja luokan *release* ilmentymä koostuu luokan *recording* olioista. Palvelurajapinta tarjoaa tietokannan oliomalliin pohjautuen joukon hakuluokkia eli URL-osoitteita, joiden avulla kunkin luokan resursseihin pääsee käsiksi. Palvelurajapinnan osoite on muotoa <http://musicbrainz.org/ws/2/>. Osoitteen verkkotunnuksen *musicbrainz.org* perässä oleva */ws/2/* kertoo, että käytetään palvelurajapinnan versiota 2, joka julkaistiin toukokuussa 2011 (*MusicBrainz Timeline – Next Generation Schema*, s.a.). Tässä työssä perehdytään versioon 2. Taulukossa 3.4 on esitetty palvelurajapinnan tarjoamat tiedonhakutavat ja GET-pyyntöjen muodot. GET-pyyntö liitetään palvelurajapinnan osoitteeseen versionumeron jäl-

²<http://lucene.apache.org/java/docs/index.html>

keen. Suorassa haussa saatavilla täytyy olla haettavan resurssin *MBID*-tunnus, kuten myös suoran haun erikoistapauksessa, selaushaussa. Indeksoitu haku ei *MBID*-tunnusta tarvitse. Indeksoidulla haulla tietoa haetaan metatietokannasta hakusanoja käyttäen.

Taulukko 3.4: MusicBrainzin palvelurajapinnan tarjoamat hakutyypit.

Hakutyyppi	GET-pyyntömuoto
suora	<i>luokka/MBID?inc=inc</i>
selaus	<i>luokka?luokka=MBID&limit=limit&offset=offset&inc=inc</i>
indeksoitu	<i>luokka?query=query&limit=limit&offset=offset</i>

GET-pyyntö alkaa luokan nimellä. Tämän jälkeen pyyntöön liitetään GET-parametrit. Taulukossa 3.5 on esitetty palvelun tarjoamat luokat. Pääluokat on esitetty lihavoituna.

Taulukko 3.5: MusicBrainz-palvelun hakuluokat

Luokka	Kuvaus
artist	Artisti on jokin ryhmä tai henkilö, joka tekee musiikkia.
release-group	Julkaisukokonaisuus on jokin tietty albumi, josta on tehty eri tyyppisiä julkaisuja, esimerkiksi eri markkina-alueille suunnattuja tai bonuskappaleita sisältäviä.
release	Julkaisu on jokin musiikillinen kokonaisuus, joka on julkaisun fyysinen ilmentymä.
recording	Recording on jollekin medialle tallennettu ääntä sisältävän työn (<i>work</i>) ilmentymä, sillä on jokin tietty kesto (<i>duration</i>) ja otsikko, ja se kuuluu aina jollekin artistille tai joillekin artisteille.
label	Levy-yhtiön merkki, joka albumin on julkaissut.
work	Edustaa musiikkiin liittyvää työtä, kuten esimerkiksi sävellystä. Työn ilmentymiä ovat kappaleet (<i>recording</i>).
tag	Folksonomiat, joiden avulla kuvaillaan tietoa.
rating	Arvostelut.
collection	Sisältää käyttäjän luoman albumikokoelman.

Albumikuuntelun tunnistamisen kannalta tärkeimpiä MusicBrainzin palvelurajapinnan tarjoamia hakuluokkia ovat *recording* ja *release*. Luokasta *recording* saadaan

haettua tietoa kappaleista hakusanojen avulla käyttäen indeksoitua hakua, kuten esimerkiksi julkaisujen nimet, joilla kappale esiintyy ja julkaisujen yksilölliset tunnisteet (*MBID*). Tunnisteiden *MBID* avulla, suoraa hakua käyttäen, julkaisujen tiedot saadaan kokonaisuudessaan hakuluokasta *release*.

Taulukossa 3.6 on esitetty palvelurajapinnan tarjoamat GET-parametrit, jotka ovat yhteisiä kaikille hakuluokille. Parametreista tärkeimpiä ovat *query*, *inc*, *limit* ja *offset*. Parametrin *query* avulla pystytään tekemään hakusanoista koostuva kysely *recording*-luokkaan. Luokasta saadaan kuunneltavalle kappaleelle esimerkiksi albumi. Parametrilla *inc* saadaan haettua albumille kappaleista käyttäen suoraa *release*-hakua. Kappalelistan lisäksi saadaan muun muassa tietoa julkaisun sisältämien medioiden määrästä. Hakujen *recording* ja *release* palauttamia tietoja käsitellään tarkemmin seuraavassa alakohdassa 3.2.2. Parametrilla *limit* voidaan rajoittaa sitä, kuinka paljon yksi pyyntö palauttaa tuloksia kerralla. Parametrin *offset* avulla on mahdollista hakea loput haun antamat tulokset, jos tuloksia on enemmän kuin yhden pyynnön vastaukseen voi mahtua.

Taulukko 3.6: Palvelurajapinnan tarjoamat GET-parametrit

Parametri	Hakutyyppi	Kuvaus
query	indeksoitu	Hakukysely, jolla tietoa haetaan hakupalvelimelta. Hakuluokkien omat hakuparametrit jätetään huomioimatta jos parametriä <i>query</i> käytetään.
inc	suora ja selaus	Parametrilla kerrotaan mitä lisätietoja resurssista halutaan. Ominaisuudet erotetaan plus-merkillä. Ilman parametriä <i>inc</i> palautetaan vain resurssin perustiedot.
limit	selaus ja indeksoitu	Palautettavien hakutulosten maksimimäärä. Oletus arvo on 25 ja suurin sallittu on 100.
offset	selaus ja indeksoitu	Palauttaa haun tulokset annetusta offset-arvosta lähtien. Arvo alkaa nolasta.
format	indeksoitu	Palautettavan tiedon esitystapa. Vaihtoehdot ovat XML ja JSON
type	indeksoitu	Parametrin avulla voi valita hakuluokan.
version	indeksoitu	MMD-formaatin versio. Oletus on 2, versio 1 on tarjolla vanhaa rajapintaa käyttäville sovelluksille.

Suoran *recording*-haun pyynnön palauttamaan XML-dokumenttiin on mahdollista liittää erilaista oheistietoa parametrin *inc* avulla. Mahdolliset parametrin *inc* arvot ovat esitetty taulukossa 3.7.

Taulukko 3.7: Suoran julkaisuhaun *release* parametrin *inc* arvot

Arvo	Kuvaus
artists	Liittää tiedon julkaisun tehneistä artisteista.
release-groups	Ryhmä, johon julkaisu kuuluu.
labels	Levymerkki, jonka alaisuudessa levyn on julkaistu.
recordings	Liittää julkaisuun kappalelistan.
media	Liittää medioiden tyypit, kuten esimerkiksi CD, DVD, Vinyl...

Julkaisun kappalelista ei esimerkiksi tule oletuksena *release*-haussa. Se saadaan sisällytettyä julkaisuun antamalla *inc* parametrille arvo *recording*. Myös jos halutaan liittää julkaisun tehneen artistin nimi, parametrin arvoksi lisätään arvo *artists*. Useat parametrin arvot erotetaan toisistaan plus-merkillä.

Taulukossa 3.8 on esitettynä indeksoidussa kappalehaussa käytettävissä olevat hakukentät. Kentistä ja niitä yhdistävistä operaattoreista, kuten AND, muodostetaan lauseke eli kysely, joka annetaan GET-parametrin *query* arvona lucene-hakumoottorille. Kyselyn lausekkeessa kentälle asetetaan arvo seuraavasti: **kenttä:"arvo"**. (*Apache Lucene - Query Parser Syntax*, 2011)

Taulukko 3.8: Indeksoidun kappalehaun *recording* parametrin *query* kentät.

Kenttä	Kuvaus
recording	Kappaleen nimi.
artist	Kappaleen artistin nimi.
release	Julkaisu, jolta kappale on.
tracks	Median sisältämien kappaleiden määrä.
tracksrelease	Julkaisun sisältämien kappaleiden kokonaismäärä.
type	Julkaisun tyyppi, joka voi saada arvon: <i>album</i> , <i>single</i> , <i>EP</i> , <i>compilation</i> , <i>soundtrack</i> , <i>spokenwords</i> , <i>interview</i> , <i>audiobook</i> , <i>live</i> , <i>remix</i> , <i>other</i> .
status	Julkaisun status, joka voi saada arvon: <i>official</i> , <i>promotion</i> , <i>bootleg</i> , <i>pseudo-release</i> .
tag	Kappaleen folksonomia.
dur	Kappaleen kesto millisekuntteina.
qdur	Kvantisoitu kappaleen kesto ($\frac{dur}{2000}$).
tnum	Kappaleen numero medialla.
position	Median numero, jolta kappale löytyy. Ensimmäinen media on numero 1.
rid	Kappaleen MBID-tunnus.
reid	Julkaisun, jolla kappale esiintyy, MBID-tunnus.
arid	Artistin, jonka kappale on, MBID-tunnus.
isrc	Kappaleen International Standard Recording Code -tunnus.

Indeksoidun *recording*-haun hyödyllisimmät kentät työn kannalta ovat *artist* ja *status*. Kyseisen haun avulla saatavat kappale tiedot sisältävät muun muassa julkaisujen nimet, ja niiden *MBID*-tunnisteet.

3.2.2 Esimerkki: albumin kappalelistan hakeminen

Tässä esimerkissä näytetään, miten albumin kappaleista saadaan haettua, kun kuunneltavasta albumista on saatavilla vain kappale ja artisti. Esimerkissä ei käytetä Last.fm-esimerkissä (3.1.2) käytettyä kappaletta, koska MusicBrainzissa ei ole artistin *Penniless* uusinta albumia *A Cab To the City*. Sen sijaan edellisen luvun esimerkissä (2.2.1) ollutta Nirvanan albumia *Nevermind* voidaan käyttää, koska se löytyy MusicBrainzista.

Otetaan tarkastelun kohteeksi albumin ensimmäinen kappale *Smells Like a Teen Spirit*. Kappaleen nimen ja artistin avulla voidaan hakea albumin nimi ja kappaleista. Julkaisut, joilla kappale esiintyy, saadaan haettua indeksoidulla *recording*-haulla seuraavasti:

```
http://musicbrainz.org/ws/2/recording?query=Smells%20Like%20Teen%
20Spirit%20AND%20artist:"Nirvana"%20AND%20status:"official"
```

Kappaleen nimen lisäksi kyselylle *query* annetaan tarkentavat kentät *artist* ja *status*. Kentän *status* avulla rajataan pois kaikki epäviralliset julkaisut, joilla kappale esiintyy, kuten esimerkiksi bootleg-tyyppiset julkaisut. (*What is a Bootleg*, s.a.) Listauksesta 3.5 nähdään, että annetuilla hakutermeillä löytyi kappaleita yhteensä 21. Tämä nähdään elementin *recording-list* attribuutista *count* (rivi 1). Kappaleiden tiedot ovat elementeissä *recording*. Elementti *recording* sisältää paljon hyödyllistä tietoa liittyen albumikuuntelun tunnistamiseen. Suoraan elementin *recording* lapsena olevassa *title*-elementissä on kappaleen nimi (rivi 4). Elementti *artist-credits* sisältää tiedon kappaleen tehneistä artisteista. Tässä tapauksessa artisteja on vain yksi (rivi 9).

Listaus 3.5: Indeksoidun *recording*-haun palauttamaa XML-merkkausta.

```
1 <recording-list offset="0" count="21">
2   <recording ext:score="100" id="5fb524f1-8cc8-4c04-a921-
   e34c0a911ea7">...</recording>
3   <recording ext:score="100" id="6da76448-982a-4a01-b65b-9
   a710301c9c9">
4     <title>Smells Like Teen Spirit</title>
5     <length>301106</length>
6     <artist-credit>
7       <name-credit>
8         <artist id="5b11f4ce-a62d-471e-81fc-a69a8278c7da">
9           <name>Nirvana</name>
```



```

10         <sort-name>Nirvana</sort-name>
11         <disambiguation>90s US grunge band</disambiguation>
12     </artist>
13 </name-credit>
14 </artist-credit>
15 <release-list>
16     <release id="4cbf7629-4695-3885-9aca-a639e62ba4ab">
17         <title>Nevermind</title>
18         <status>Official</status>
19         <release-group type="Album"/>
20         <date>1991</date>
21         <medium-list>
22             <track-count>12</track-count>
23             <medium>
24                 <position>1</position>
25                 <track-list offset="0" count="12">
26                     <track>
27                         <title>Smells Like Teen Spirit</title>
28                     </track>
29                 </track-list>
30             </medium>
31         </medium-list>
32     </release>
33 </release-list>
34 </recording>
35 </recording-list>

```

Edellä mainittujen tietojen lisäksi *recording* sisältää julkaisujen tiedot, joilla kappale esiintyy. Yllä olevassa listauksessa näkyvä 301106 millisekuntia kestävä *Smells Like a Teen Spirit* -kappale esiintyy vain yhdellä julkaisulla, *Nevermind*. Kyseisen julkaisun tiedot ovat elementin *release-list* sisällä olevassa elementistä *release* (alkaa riviltä 16 ja loppuu riville 32). Elementin *release* sisällä olevia, julkaisua kuvaavia lapsi-elementtejä ovat *title*, *status*, *release-group*, *date* ja *medium-list*. Elementti *title* sisältää julkaisun nimen (rivi 17), *status* ilmaisee julkaisun virallisuuden (rivi 18). Elementin *release-group* attribuutti *type* sisältää julkaisun tyyppin (rivi 19) ja elementti *date* sisältää julkaisuvuoden (rivi 20). Elementin *medium-list* sisältämistä tiedoista oleelliset ovat julkaisun kappalemäärä (rivi 22), kappaleen sijainti medialla (rivi 24) ja median kappalemäärän (rivi 25).

Kappaleet (*recording*) ovat järjestetty yllä olevassa listauksessa 3.5 laskevaan järjestykseen pisteytysarvon (*ext:score*) ja kappaleen keston (*length*) mukaan. Elementin *recording* sisältämien julkaisujen (*release*) järjestyksestä ei ole havaittavissa vastaavaa johdonmukaisuutta, kuin kappaleilla on havaittavissa.

Listauksesta 3.5 saadaan kappalelistan hakemisessa tarvittava julkaisun *MBID*-tunnus, joka on elementin *release* attribuutissa *id* (rivi 16). Julkaisun tiedot kap-

paleineen saadaan suoralla *release*-haulla käyttämällä *MBID*-tunnusta. Haun GET-pyynnön osoite on seuraavanlainen:

<http://musicbrainz.org/ws/2/release/4cbf7629-4695-3885-9aca-a639e62ba4ab?inc=recordings>

Parametrin *inc* arvo "recording" liittää pyynnön palauttamaan XML-dokumenttiin kappalelistan. Osa saadusta XML-tiedosta on esitetty listauksessa 3.6. Julkaisun kappaleet ovat lueteltu *track-list*-elementin sisällä (riviltä 12 riville 21). Elementin *track-list* attribuutista *count* nähdään julkaisun kappaleiden kokonaismäärä, jonka arvo on 12 (rivi 12). Elementti *track-list* on elementin *medium* lapsi. Elementin *medium* avulla julkaisun kappaleet ryhmitellään tallennusmedioittain. Tässä tapauksessa julkaisu koostuu vain yhdestä mediasta, joka nähdään elementin *medium-list* attribuutin *count* arvosta (rivi 9).

Listaus 3.6: Suoran *release*-haun palauttamaa XML-merkkausta.

```

1 <release id="4cbf7629-4695-3885-9aca-a639e62ba4ab">
2   <title>Nevermind</title>
3   <status>Official</status>
4   <quality>normal</quality>
5   <text-representation>...</text-representation>
6   <date>1991</date>
7   <country>CA</country>
8   <barcode>77749749200602</barcode>
9   <medium-list count="1">
10     <medium>
11       <position>1</position>
12       <track-list count="12">
13         <track>
14           <position>1</position>
15           <recording id="6da76448-982a-4a01-b65b-9a710301c9c9">
16             <title>Smells Like Teen Spirit</title>
17             <length>301106</length>
18           </recording>
19         </track>
20         ...
21       </track-list>
22     </medium>
23   </medium-list>
24 </release>

```

4. STATSTER

Statster on verkkopalvelu, joka taltioi pienen osan elämää — pienen, mutta tärkeän. Se taltioi kuunnellut musiikkialbumit, jotka koostuvat kappaleista. Yksittäisiä kappaleita ei taltioida. Poikkeuksia kuitenkin on, kuten esimerkiksi yhtyeen *Green Carnation* albumi *Light of Day, Day of Darkness* (*Light of Day, Day of Darkness*, s.a.). Kyseinen albumi sisältää vain yhden kappaleen. Tästä syystä sen kuunteleminen on albumikuuntelu ja se voidaan lisätä Statsteriin. Yksittäisten kappaleiden taltioimiseen on jo olemassa useampi verkkopalvelu, kuten Last.fm ja Libre.fm¹, joka on tehty vastalauseena Last.fm:n kaupallisuudelle (*Libre.fm About*, s.a.).

Tiedon keräämisen lisäksi Statsterin yksi tärkeimmistä tehtävistä on koota tiedon palasista eli yksittäisistä albumikuunteluista loogisia kokonaisuuksia. Kokonaisuudet voivat olla esimerkiksi tietyn albumin kuuntelumäärät halutulta ajalta, koko kuunteluhistorian kuunnelluin albumi tai viimeisimmät albumikuuntelut. Näitä moninaisia tilastoja tullaan käsittelemään myöhemmin tässä luvussa. Aluksi perehdytään palvelun käyttöön ja rakenteeseen.

4.1 Käytön aloittaminen

Statster pohjautuu käyttäjien luomaan sisältöön, joten palveluun täytyy rekisteröityä ennen käytön aloittamista. Käyttäjätunnuksen luominen onnistuu rekisteröitymissivulta. Sivulle pääsee etusivun kautta tai sivuston yläosan hallintapalkin ”Rekisteröidy” -linkistä. Hallintapalkki on nähtävillä kuvassa 4.1. Käyttäjältä kysytään rekisteröitymisen yhteydessä vain käyttäjänimi ja salasana sekä ihmisyyssvarmenne. Ihmisyyssvarmenteella pyritään varmistamaan, että käyttäjä on ihminen. Tarkempien tietojen antaminen on käyttäjän vapaaehtoisesti päätettävissä.

Tarkempien tietojen lisääminen onnistuu *Muokkaa profiilia* -sivulta. Sivulle pääsee joko hallintapalkin valikosta tai omalta profiilisivulta. Hallintapalkin valikon saa auki oman nimen vieressä olevasta alaspäin osoittavasta nuolesta. Omalle profiilisivulle pääsee nuolen vieressä olevasta käyttäjänimestä. Hyödyllisimmät tiedot, jotka käyttäjän on hyvä lisätä rekisteröitymisen jälkeen, ovat: sähköpostiosoite, oma kuva ja Last.fm-käyttäjätunnus jos sellainen on. Sähköpostiosoitteen avulla käyttäjää voidaan informoida palvelussa tapahtuvista asioista, ja sitä voidaan hyödyntää salasanan ja käyttäjätunnuksen palauttamisessa. Käyttäjäkuvan avulla käyttäjä erot-

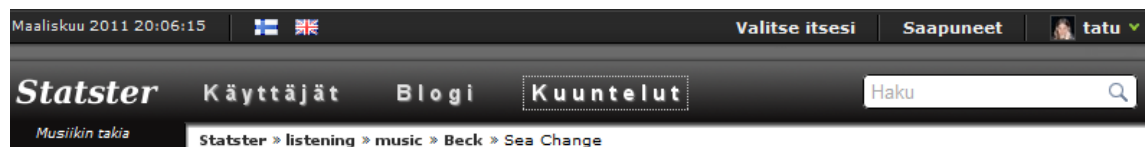
¹<http://libre.fm/>

tuu helpommin palvelun muiden käyttäjien joukosta. Kuvan voi lisätä joko omalta koneelta tai verkkosijainnista URL-osoitteen avulla. Last.fm-tunnuksen antaminen mahdollistaa vaihtoehtoisen kuuntelunlisäystavan, jonka tarkoitus on tehdä verkkopalvelun kautta tapahtuvasta kuuntelun lisäämisestä helpompaa. Kuuntelun lisääminen ilman Last.fm-tunnusta onnistuu joko lomakkeen avulla tai oheisohjelmalla suoraan musiikkisoittimesta.

Statsteriin lisätyt kuuntelut ovat julkisesti nähtävillä kaikille Internetin käyttäjille. Tällä hetkellä niistä ei ole mahdollista tehdä yksityisiä. Kirjautuneen ja kirjautumattoman käyttäjän ero on siinä, että kirjautumaton käyttäjä ei pysty vaikuttamaan palvelun sisältöön mitenkään. Kirjautunut käyttäjä voi muun muassa lisätä itselleen kuunteluita, lisätä oheistietoa albumeihin ja artisteihin, lähettää kavereille albumisuosituksia ja kirjoittaa blogia. Käyttäjä voi myös poistaa kaiken lisäämänsä tiedon. Sisään kirjautuneen käyttäjän on myös helppo suodattaa sivujen sisältö näyttämään vain tietoa omista kuunteluista. Itsensä valitseminen onnistuu sivun yläosassa olevasta linkistä ”Valitse itsesi”. Muiden käyttäjien valitseminen onnistuu hakua käyttämällä. Hakukenttä ja ”Valitse itsesi” -linkki näkyvät kuvassa 4.1 oikealla puolella.

4.2 Sivuston pääosiot

Statster koostuu kolmesta pääosiosta, jos etusivua ei lasketa, ne ovat: *Käyttäjät*, *Blogi* ja *Kuuntelut*. Osiot muodostavat sivuston päänavigaation, joka on nähtävillä kuvassa 4.1.



Kuva 4.1: Sivuston päänavigaatio, joka on yhteinen kaikille sivuille.

Käyttäjät-osion pääsivulla on tietoa palvelun käyttäjistä ja erilaisia listauksia, joita virroiksi myös kutsutaan. Virroissa on tietoa käyttäjien viimeisimmistä toimista, kuten albumien tykkäyksistä, artistien fanituksista, blogikirjoituksista ja palveluun jätetyistä kommentteista. Sivulta pääsee yksittäisen käyttäjän profiilisivulle, jossa on tietoa käyttäjästä ja koostetusti tietoa: käyttäjän kuunteluhistoriasta, tykkäyksistä, fanituksista ja blogikirjoituksista. Omalta profiilisivulta pääsee myös muokkaamaan profiiliaan.

Blogi-osiossa käyttäjät voivat kirjoittaa omia tekstejä ja kommentoida muiden käyttäjien kirjoituksia. Blogissa yksittäisen käyttäjän kirjoitukset saadaan näkyviin haun avulla kirjoittamalla käyttäjän nimi ja valitsemalla se. Haku on nähtävillä yllä olevassa kuvassa 4.1 oikealla. Sivujen sisältämä tieto on myös mahdollista suodattaa

samalla tavalla muissa osioissa, kuten *Kuuntelu*-osiossa.

4.3 Kuuntelut-osio

Kuuntelut-osio, nimensä mukaisesti, pitää sisällään kaiken oleellisen tiedon albumikuunteluihin liittyen. Osion pääsivu kokoaa palveluun lisätyistä viimeisimmistä kuunteluista havainnollisia kokonaisuuksia. Kokonaisuudet ovat tilastoja kuuntelumääristä käyttäjittäin, albumeittain ja artisteittain. Albumien kuuntelumääristä käyttäjittäin esitetään tietoa kuvaajassa, jossa on pylväiden avulla havainnollistetaan kuuden eniten kuunnelleen käyttäjän kuuntelumäärät viimeisen kuukauden (31 päivän) ajalta. Albumien ja artistien kuuntelumäärät ovat esitetty viiden kuunnelluimman listana. Alla olevassa kuvassa 4.2 on nähtävillä viisi kuunnelluinta albumia viimeisen kuukauden ajalta. *Kuunnelluimmat artistit* -lista on muuten samanlainen, paitsi siinä näytetään artistit, joiden albumeita on kuunneltu eniten valitulla aikavälillä. Aikavälit ovat kiinteiksi asetettu, ja ne ovat viimeisin viikko, kuukausi, 3kk, 6kk, 12kk ja yhteensä, joka sisältää kaikki palvelun koko olemassa olon ajalta.





































Kuva 4.2: Viisi kuunnelluinta albumia viimeisen kuukauden ajalta suodatettuna käyttäjällä *tatu*.

Kuuntelut-osion pääsivulla on myös lista kahdeksasta palveluun viimeksi lisätyistä albumikuuntelusta. Tätä näkymää kutsutaan kuunteluvirraksi. Se on nähtävillä kuvassa 4.3.

Kuunteluvirrasta nähdään tietty määrä viimeksi lisättyjä albumikuunteluita. Määrä ja sisältö riippuu siitä, millä sivulla virta on käytössä. *Kuuntelut*-osion pääsivulla näytetään vähemmän mitä *Viimeisimmät kuuntelut* -sivuilla. Virran avulla myös näytetään tietyn artistin viimeisimmät albumikuuntelu ja tietyn albumin viimeisimmät kuuntelut. Sivuille pääsee artisti- ja albumisivujen kautta. Virran sisältö on mahdollista rajata käyttäjäkohtaiseksi. Tällöin virran käyttäjäkuvat poistuvat. Käyttäjäkuvat ovat näkyvissä kuvan 4.3 oikeassa reunassa. Kuunteluvirta-kuvassa

oleva sydän tarkoittaa, että käyttäjä tykkää albumista. Kuvan tapauksessa artistin *Penniless* albumista *A Cab To The City*. Pienet pyöreät pallot virrassa tekstien “*artisti – albumi (julkaisuvuosi)*” perässä ovat kuuntelun poistamista varten. Ne näkyvät vain käyttäjän ollessa sisäänkirjautuneena niiden kuuntelujen kohdalla, jotka ovat hänen kuuntelemaiaan.

päivitetty 14:42

	Penniless – A Cab To The City (2010) 			 juuri äsken	
	Elliott Smith – Roman Candle (1994) 			tänään	
	PJ Harvey – To Bring You My Love (1995) 			tänään	
	Thee Silver Mt. Zion Memorial ... – 13 Blues For Thirteen Moons (2008) 			tänään	
	HIM – Razorblade Romance (1999) 			eilen	
	Rush – Roll The Bones (1991) 			eilen	
	New Found Glory – Not Without A Fight (2009) 			eilen	
	Porcupine Tree – Warszawa (2001) 			eilen	

Kuva 4.3: Viimeisimmät kuuntelut -virta, jossa on näkyvillä kahdeksan viimeisintä kuuntelua.

4.3.1 Artisti- ja albumisivut

Artisti- ja albumisivuille on mahdollista päästä useita eri reittejä pitkin, kuten esimerkiksi *Kuuntelut*-osion pääsivulla olevista artisti- ja albumi-linkeistä tai artisti- ja albumikuvia painamalla. Artistin tai albumin nimen ollessa tiedossa sivulle pääsee helpoiten hakua käyttämällä. Hausta on tehty käyttäjäystävällinen. Se näyttää käyttäjälle hakutermiä kirjoitettaessa hakutuloksia. Tulokset aukeavat listana hakukentän alle. Lista on nähtävillä alla olevassa kuvassa 4.4. Esimerkiksi jos listalta valitaan vaikkapa artisti *Pink Floyd*, siirrytään suoraan kyseisen artistin sivulle.



Kuva 4.4: Haun ehdotuslista, josta on mahdollista valita hakutermi.

Artisti- ja albumisivut eroavat toisistaan hyvin vähän. Ne koostuvat samoista sisältöelementeistä. Erona on se, että artistisivulla näytettävät kuuntelutilastot koostaan kaikista artistin albumeista, albumisivulla näytetään vain valittuna olevasta albumista.

Kuvassa 4.5 on nähtävillä osa albumisivun yläosaa artistin *Beck* albumista *Sea Change*. Kuvassa näkyy vasemmalla alhaalla otsikkoteksti ”Taggaa Tätä”, jonka alla on lomake. Lomakkeen avulla albumiin voidaan liittää oheistietoa (metatietoa). Sen avulla voidaan lisätä neljän tyyppistä tietoa: tyylilajeja, avainsanoja, kansallisuuksia ja artistiin liittyviä artisteja.

Sea Change (2002)

Albumi lisätty Statsteriin vuonna viime helmikuussa



10 kuuntelua (#666)

10 kuuntelua kirjastossasi (#66)

Tyylilajit: **Alternative Rock** **Folk Rock** **Singer/Songwriter**

Avainsanat: **2000's**

Muuta albumin kantta

Albumin tykkäykset:

Biografia **Kuuntelut Kuukausittain** **Tilastot**

The Artist name Beck can refer to two artists; 1) Beck Hansen (born Bek David Campbell, July 8, 1970) is an American musician, singer-songwriter, and multi-instrumentalist. Beck's music is known for its pop-junk culture collage of musical styles, ironic lyrics, samples, drum machines, live instrumentation and heady sound effects. Beck was born in Los Angeles, California, to David Campbell (a musician and son of a Presbyterian minister) and Bibbe Hansen (a former dancer for The Velvet Underground, founding member of the satirical band Black Fag, and visual artist).

[Katso lisää](#)

Taggaa Tätä

anna albumille erilaista lisätietoa

Valitse tyyppi

Erottele useat arvot pilkuilla. Lisää vain yksi artisti kerrallaan.

Lähetä metatieto

Viimeisimmät Kuuntelut

albumin viimeisimmät kuuntelut

 eilen

 eilen

 5 päivää sitten

Kuva 4.5: Albumisivun yläosa, joka sisältää tietoa albumista sekä mahdollistaa metatiedon lisäämisen.

Albumisivulta tapahtuu myös albumin suosikkeihin lisäys eli tykkääminen. Kuvassa oikealla ylhäällä on nähtävillä sydän, jota painamalla käyttäjä voi merkitä albumin tykätyksi. Tykkäysten määrä näkyy albumikuvan alla sydämenä. Albumin sisältäessä useampia tykkäyksiä sydämen viereen ilmestyy tykkäysten lukumäärä. Artisteja on mahdollista fanittaa samalla tavalla kuin albumeita tykätä.

Albumikuvan oikealla puolella on elementti, joka sisältää kolme välilehteä. Kuvassa 4.5 aktiivisena on *Biografia*-välilehti, jossa näytetään lyhyt kuvaus artistista vaikka oltaisiin albumisivulla. Kuvaus haetaan Last.fm-palvelusta. Kaksi muuta välilehteä sisältävät tilastoja valitun albumin tai artistin kuunteluista.

Biografian alla on nähtävillä lista albumin viimeisimmistä kuunteluista. Otsikkoa “Viimeisimmät kuuntelut” tai listan alla olevaa “Katso lisää” -linkkiä painamalla päästään kuvan 4.3 tyyppiseen kuunteluvirta-näkymään, jossa näytetään kaikki valitun albumin viimeisimmät kuuntelut.


Kuvassa 4.6 on nähtävillä albumisivun keskiosaa, jossa vasemmalla puolella on artistin *Beck* albumit. Albumit ovat järjestetty kuuntelumäärien mukaan. Tämä “Suosituimmat Albumit” -näkyvä on sama, oli kyseessä sitten albumi- tai artistisivu.

Suosituimmat Albumit

artistilta Beck


Yhteensä Kirjastossasi

Julkaistu: 2002




Sea Change
18 kuuntelua

Julkaistu: 1998




Mutations
6 kuuntelua

Julkaistu: 1994




One Foot In The Grave
3 kuuntelua

Julkaistu: 2008




Modern Guilt
2 kuuntelua

Julkaistu: 1999




Julkaistu: 1994



Eniten Kuunnelleet

albumia eniten kuunnelleet







tatu (18)



[Katso lisää](#)

Sisällöt

sisältäen tietoa tästä artistista.





[PROG ARCHIVES](#)

[Rate Your Music](#)

Kiertueella

listataan artistin 15 tulevaa tapahtumaa



Los Angeles, United States

Kuva 4.6: Albumisivun keskiosa.

Kuvan 4.6 oikeanpuoleisessa osassa ylhäältä aloitettuna on näkyvillä lista, jossa on listattu albumia eniten kuunnelleet käyttäjät. Listassa näytetään enintään kuusi käyttäjää. Linkistä “Katso lisää” päästään sivulle, jossa on nähtävillä kaikkien käyttäjien kuuntelumäärät. Tiedot on esitetty vaakatasoon asetettujen pylväiden avulla. Kuvassa 4.8 on nähtävillä tämän tyyppinen kuuntelumäärien esitysmuoto, mutta eri sisällöllä.







Eniten Kuunnelleet -näkyvän alla on linkkejä palvelun ulkopuolisiin sivustoihin, jotka tarjoavat lisää tietoa tarkasteltavasta sisällöstä. Esimerkiksi “Spotify”-kuvaa painettaessa saadaan aukaistua soittimeen Spotify artistin *Beck* albumi *Sea Change*. Kuva on esitetty harmaana, jos albumi ei ole kuunneltavissa Spotifyssä. Kaikki muut linkit ovat aina samannäköisiä.

Alimpana oikeassa reunassa näkyvillä on lista valitun albumin artistin tulevista keikoista. Painettaessa kiertuelinkkiä siirrytään sivun alaosassa olevalle kartalle, jonka avulla kiertuepaikkojen sijainnit esitetään havainnollisesti. Karttana käytetään Googlen tarjoamaa sulautettavaa karttaa².

4.3.2 Suosituimmat albumit ja artistit

Suosituimmat-näkymät ovat albumien kuuntelumäärien pohjalta luotuja tilastoja. Näiden näkymien avulla näytetään Statsterin kuunnelluimmat albumit ja artistit. Sisältöä on mahdollista rajata erilaisilla ominaisuuksilla, kuten ajanjaksolla ja käyttäjällä. *Albumi*- ja *Artisti*-näkymät eroavat toisistaan vain siten, että *Suosituimmat artistit* -näkymän sisältö on koottu artistien kaikkien albumien kuuntelumäärien pohjalta. *Suosituimmat*-näkymät voidaan jakaa kolmeen kategoriaan sen perusteella, miten tieto niissä esitetään: suosituimmat ajanjaksoittain, pylväin ja isoin kuvin. Tietyn kategorian näkymässä voidaan esitetään tilastoja joko artistien tai albumien kuuntelumääristä.

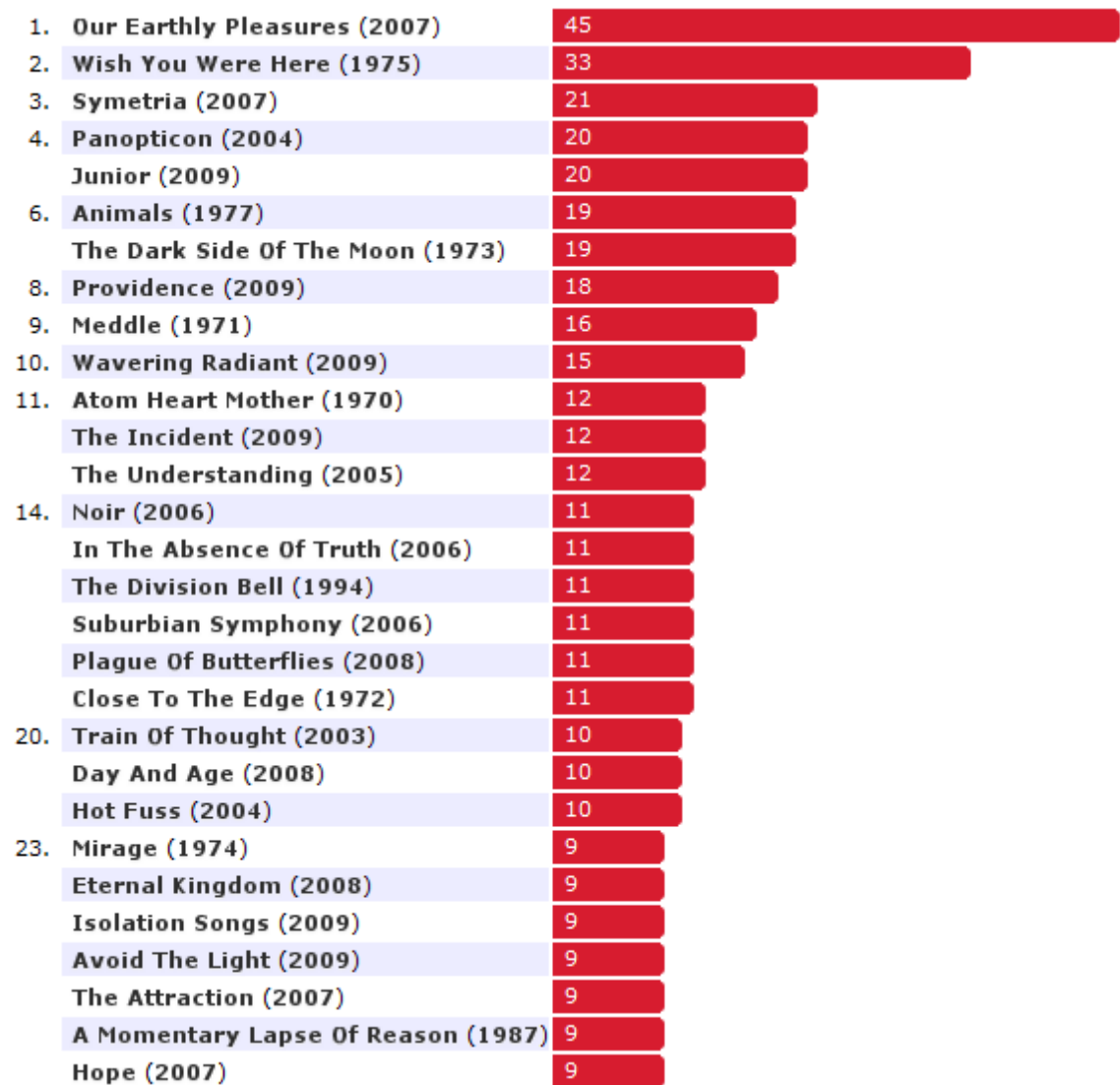
Kuvassa 4.7 on nähtävillä kategorian ”suosituimmat ajanjaksoittain” näkymä. Näkymän sisältö on rajattu käyttäjällä *tatu*. Näkyvillä on vuosien 2010 ja 2009 kolme kuunnelluinta albumia. Näkymässä olevia ”Katso lisää” -linkkejä painamalla pääsee sivulle, jossa on nähtävillä loput albumit.

2010			
1.		Pink Floyd – Animals (1977)	34 kuuntelua
2.		Herra Ylppö Ja Ihmiset – Pojat Ei Tanssi (2010)	21 kuuntelua
		Pink Floyd – Wish You Were Here (1975)	21 kuuntelua
Katso lisää ➔			
2009			
1.		Maximo Park – Our Earthly Pleasures (2007)	45 kuuntelua
2.		Pink Floyd – Wish You Were Here (1975)	33 kuuntelua
3.		Hidria Spacefolk – Symetria (2007)	21 kuuntelua
Katso lisää ➔			

Kuva 4.7: Suosituimmat albumit vuosittain suodatettuna käyttäjällä *tatu*.

²<http://maps.google.com/>

Vuoden 2009 “Katso lisää” -linkkiä painamalla päästään kuvan 4.8 mukaiseen näkymään, joka kuuluu “suosituimmat pylväin” -kategoriaan. Alla olevassa kuvassa on näkyvillä vain ensimmäiset 20 albumia havainnollisesti vaakatasossa olevin pylväin esitettynä.



Kuva 4.8: Suosituimmat albumit vuodelta 2009 suodatettuna käyttäjällä *tatu*.

Kategorian “suosituimmat isoin kuvin” mukaisissa tilastoissa kuuntelumäärät ovat koottu aikaväliltä, joka alkaa tietystä ennalta määritetystä menneestä ajankohdasta ja päättyy aina nykyhetkeen. Aikavälit ovat seuraavat: viimeinen viikko, kuukausi, 3kk, 6kk, 12kk ja kaikki palvelun koko olemassa olon ajalta. Kyseisen kategorian tilastoja on nähtävillä aiemmin esitetyissä kuvissa 4.2 ja 4.6.

4.3.3 Suosikit
















Statsterissa on mahdollista luoda suosikkilistoja artisti- ja albumisivujen kautta. Artistista tykkäämistä eli fanitusta symboloi tähti ja albumista tykkäämistä sydän. Kuvassa 4.9 on nähtävillä *Suosikit*-sivun pääsivu, joka kokoaa kaikki suosikit listan muodossa. Lista on järjestetty tykkäyspäivämäärän mukaan laskevaan järjestykseen uusimmasta vanhimpaan.

Yhteensä **75** tykättyä albumia ja fanitettua artistia.

♥ Näytä vain tykättyt albumit »

★ Näytä vain fanitetut artistit »

Sivu: 1 2 [Seuraava](#)

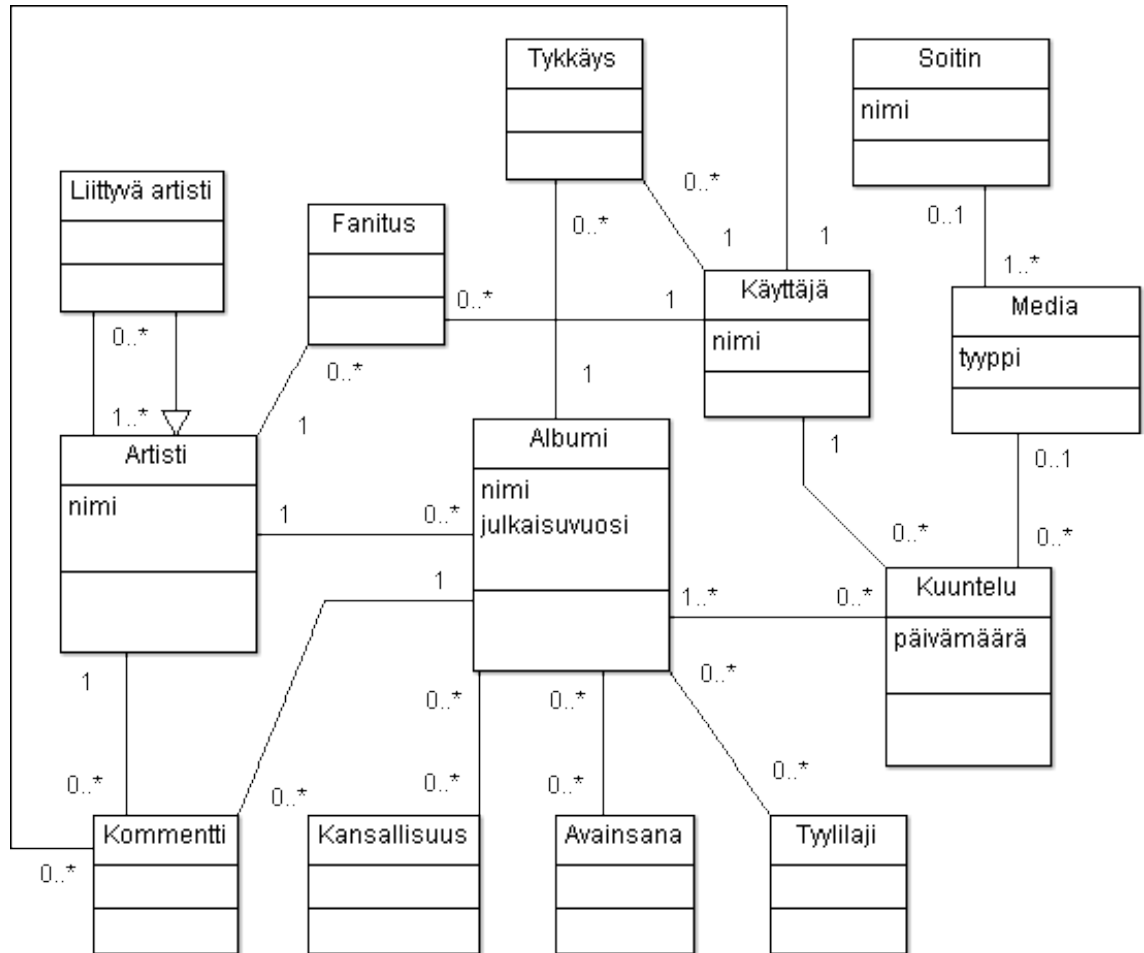
	Neil Young – After The Gold Rush (1970)	♥	yli 3 viikkoa sitten
	Neil Young And Crazy Horse – Rust Never Sleeps (1979)	♥	yli 3 viikkoa sitten
	Neil Young	★	viime toukokuussa
	Marc Bolan	★	viime toukokuussa
	Ryan Adams – Heartbreaker (2000)	♥	viime huhtikuussa
	Ryan Adams	★	viime huhtikuussa
	Beck	★	viime maaliskuussa
	Elliott Smith	★	viime maaliskuussa
	Beck – Sea Change (2002)	♥	viime maaliskuussa
	Jukka Poika – Kylmästä Lämpimään (2010)	♥	viime helmikuussa
	Damien Rice	★	viime helmikuussa
	Elton John	★	viime tammikuussa
	Creedence Clearwater Revival – Pendulum (1970)	♥	viime tammikuussa
	Penniless – A Cab To The City (2010)	♥	joulukuussa 2010
	Röyksopp – The Understanding (2005)	♥	lokakuussa 2010

Kuva 4.9: Käyttäjän *tatu* suosikki albumeita ja artisteja.

Suosikit-sivu on mahdollista suodattaa käyttäjällä, kuten yllä olevassa kuvassa on tehty. Tämän lisäksi voidaan rajata pois joko fanitukset tai tykkäykset. Kuvan linkistä “Näytä vain tykättyt albumit” saadaan fanitukset pois näkyviltä, ja päin vastoin alla olevasta linkistä “Näytä vain fanitetut artistit”.

4.4 Tietomalli

Statster käyttää relaatiotietokantaa nimeltä MySQL³ tiedon varastointiin. Kuvassa 4.10 on esitetty UML-standardin mukainen yksinkertaistettu tietomalli tietokannan sisältämien taulujen rakenteesta ja niiden välisistä suhteista.



Kuva 4.10: Yksinkertaistettu versio Statsterin tietosisällöstä luokkakaavion muodossa.

Palvelun kannalta keskeisimmät luokat, joista albumikuuntelu koostuu, ovat *Artisti*, *Albumi*, *Kuuntelu* ja *Käyttäjä*. Albumin kuuntelemiseen liittyy aina jokin tietty albumi, joka kuuluu jollekin tietylle artistille. Kuuntelun lisäämisen yhteydessä on mahdollista liittää kuunteluun tieto, miltä medialta albumi ollaan kuunneltu ja myös tieto siitä, millä soittimella albumi ollaan kuunneltu. Kuvasta 4.10 nähdään, että luokkaan *Kuuntelu* voi liittyä korkeintaan yksi *Media*. Luokkaan *Media* voi puolestaan liittyä korkeintaan yksi *Soitin*. Luokan *Kuuntelu* attribuutti *päivämäärä* sisältää vuoden, kuukauden ja päivän.

³<http://www.mysql.com/>

Albumikuuntelun lisää palveluun aina rekisteröitynyt käyttäjä. Kuvassa sitä esittää luokka *Käyttäjä*. Käyttäjän suhde kuunteluun on yksikäsitteinen eli tietty kuuntelu voi kuulua vain yhdelle käyttäjälle. Kuvasta nähdään myös, että käyttäjällä voi olla suhde luokkiin *Artisti* ja *Albumi* luokkien *Fanitus* ja *Tykkäys* kautta eli käyttäjä voi fanittaa artisteja ja tykätä albumeista. Kyseiset suhteet mahdollistavat suosikilistojen tekemisen.

Albumin sisältöä voidaan kuvailla neljän luokan avulla, jotka ovat *Kansallisuus*, *Avainsana*, *Tyylilaji* ja *Kommentti*. Luokista *Kansallisuus* on sisällöltään kaikkein rajoittunein ja *Kommentti* vapaamuotoisin. Kommentin kirjoittaa aina joku palvelun rekisteröitynyt käyttäjä omilla nimissään.

Artistia on mahdollista kuvailla vain luokilla *Kommentti* ja *Liittyvä artisti*. Artistin kommentin on kirjoittanut rekisteröitynyt käyttäjä, kuten albuminkin tapauksessa. Artistiin liitettävän liittyvän artistin pitää olla jokin palvelussa jo oleva artisti. Kyseeseen voi tulla esimerkiksi jonkun yhtyeen henkilön sooloprojektista syntynyt artisti. Kuten esimerkiksi artisti *Pink Floyd* liittyy näin artistiin *David Gilmour*.

5. ALBUMIKUUNTELUJEN LISÄÄMINEN STATSTERIIN

Statster-verkkopalvelun yksi tärkeimmistä tehtävistä on tiedon kerääminen. Palveluun on mahdollista lisätä albumikuunteluja joko verkkopalvelun kautta tai palvelurajapinnan avulla.













Tässä luvussa käsitellään aluksi kuuntelun lisäämistä verkkopalvelun kautta. Tämän jälkeen perehdytään Statsterin tarjoamaan palvelurajapintaan, joka mahdollistaa kuuntelujen lisäämisen suoraan käyttäjän soitinohjelmistosta. Lopuksi tutustutaan Statsteria varten tehtyihin oheisohjelmiin, joiden avulla kuunnellut albumit saadaan automaattisesti tunnistettua kuunnelluiksi ja lähetettyä Statsteriin.

5.1 Kuuntelun lisääminen verkkopalvelun kautta

Web-käyttöliittymä tarjoaa kaksi tapaa kuuntelun lisäämiseen: lomakkeen ja Last.fm-verkkopalveluun pohjautuvan albumivirran. Lomake on kätevä satunnaisten albumikuuntelujen lisäämisessä. Lomaketta käytettäessä kuuntelun lisääminen koostuu muutamasta vaiheesta. Ensimmäinen vaihe on päivämäärän valitseminen. Tätä ei tarvitse tehdä, jos albumi ollaan kuunneltu kyseisenä päivänä, koska päivämäärä on oletuksena asetettu nykyhetkeen. Päivämäärän jälkeen valitaan artisti, jonka albumi ollaan kuunneltu. Artistin valitsemisen jälkeen avautuu kenttä, johon syötetään kuunnellun albumin nimi. Artisti- ja albumikentissä on käytössä automaattisesti nimiä ehdottava täydennys, joka on lähestulkoon samanlainen kuin aiemmin esitetyssä kuvassa 4.4. Jos kuunneltua albumia ei vielä ole Statsterissa, kysytään sen julkaisu-vuosi. Jos albumi on jo Statsterissa, siirrytään suoraan kuuntelumedian ja soittimen valintaan. Median ja soittimen valinta on vapaaehtoinen. Tämän jälkeen kuuntelu voidaan lähettää palveluun.

Lomake ei ole kätevä jos kuunnellaan paljon albumeita, koska lomaketta käytettäessä käyttäjän tarvitsee muistaa kuunnellut albumit ja lisätä jokainen yksitellen. Kuuntelun lisäämisen helpottamiseksi palveluun on luotu vaihtoehtoinen tapa, joka hyödyntää Last.fm-palvelua. Last.fm:ää hyödyntäen käyttäjän muistamistaakkaa pystytään vähennetään. Se myös vähentää Statster-palvelussa vierailun tarvetta. Tämä vaihtoehtoinen tapa, jota kutsutaan Last.fm-albumivirraksi, on nähtävillä kuvassa 5.1. Kuunnelluista albumeista on nähtävillä kansikuvat, jotka nopeuttavat oikean albumin löytämistä listalta. Kansikuvan jälkeen on näkyvillä artistin ja albumin ni-

mi. Albumien nimet, joita ei vielä ole Statsterissa, ovat harmaana. Albumin nimen jälkeen on painike “Lisää”, jota klikkaamalla saadaan näkyviin lomake, josta albumi-kuuntelu lisätään. Lisäyspainikkeen viereen oikealle puolelle tulee näkyviin pyöreä, vihreä ikoni, kun albumille ollaan lisätty kuuntelu. Ikoni on nähtävissä listan ensimmäisessä albumissa *Sea Change* ja viimeisessä *Razorblade Romance*. Albumivirran oikeanpuoleisella sivulla on nähtävillä ajankohta, jolloin albumi ollaan kuunneltu. Edellä mainittujen lisäksi kuvassa on “Päivitä kuuntelut” -painike, josta albumivirta saadaan päivitettyä ilman koko sivun päivittämistä.

Päivitä kuuntelut			
	Beck – Sea Change	Lisää 	tänään
	So Called Plan – Suburban Symphony	Lisää	tänään
	Penniless – A Cab To the City	Lisää	tänään
	Elliott Smith – Roman Candle	Lisää	tänään
	PJ Harvey – To Bring You My Love	Lisää	tänään
	Manboy – Something Wonderful	Lisää	tänään
	A Silver Mt. Zion – 13 Blues For Thirteen Moons	Lisää	tänään
	Tool – Ænima	Lisää	eilen
	Camel – On the Road 1982	Lisää	eilen
	HIM – Razorblade Romance	Lisää 	eilen

Kuva 5.1: Lista viimeksi kuunnelluista albumeista Last.fm palvelua hyödyntäen.

Last.fm-palvelua hyödyntävä kuuntelun lisäys vaatii toimiakseen käyttäjän Last.fm-tunnuksen, eli käyttäjän täytyy luoda itselleen Last.fm-profiili, jos sellaista ei ole. Tämän lisäksi käytettävän soitinohjelmiston täytyy lähettää kappalekuuntelut Last.fm-palveluun. Toiminta perustuu Last.fm-kuunteluvirran hyödyntämiseen. Kuunteluvirta ei ole täysin albumitarkka. Se saattaa sisältää albumeita, joita ei olla kuunneltu, koska kappaleet voivat kuulua usealle eri albumille. Kyseeseen voi tulla esimerkiksi kokoelma-albumi tai live-albumi. Käyttäjän tehtäväksi jää oikeiden julkaisujen valitseminen listasta, joka näkyy kuvassa 5.1. Kuuntelu lisätään albumivirrasta painamalla halutun albumin kohdalla “Lisää”-painiketta. Tässä tapauksessa kuuntelu lisätään artistin *Manboy* albumille *Something Wonderful*. Alle aukeaa kuvan 5.2 mukainen lomake, jonka avulla on mahdollista muokata albumin nimeä ja lisätä kuuntelulle formaatti eli media, jolta albumi ollaan kuunneltu. Median valinnan jälkeen aukeaa lista, josta voi valita käytetyn soittimen.

Kuuntelu lisätään Statsteriin painamalla “Lisää kuuntelu” -painiketta. Uuden, Statsterista puuttuvan albumin yhteydessä kysytään vielä albumin julkaisuvuosi.

Kun julkaisuvuosi ollaan annettu, uusi albumi lisätään palveluun ja albumiin liitetään kuuntelu painettaessa uudesti “Lisää kuuntelu” -painiketta.

Manboy - Something Wonderful Lisää tänään

▼ **Valitse albumi**

Something Wonderful

Artistin albumit voidaan listata painamalla nuoli alaspäin painiketta (↓) tai voit luoda uuden albumin tälle artistille

▼ **Valitse kuunteluformaatti**

☒ Ei Valittu

☐ Kasetti

☐ CD-levy

☐ Vinyylilevy

☐ DVD-levy

☐ Tiedosto

☐ Virta

Jos haluat voit antaa formaatin, jossa kuuntelit albumin. Tämä valinta on vapaaehtoinen.

Lisää kuuntelu Sulje

Kuva 5.2: Painiketta “Lisää” klikattaessa avautuva kuuntelun lisäys -lomake, joka mahdollistaa albumin nimen muokkauksen ja kuuntelumedian ja soittimen valitsemisen.

Last.fm-palvelua hyödyntävä albumivirta helpottaa jonkin verran kuuntelun lisäämistä, mutta vaatii silti käyttäjältä osallistumista tiedon syöttämiseen. Käyttäjän täytyy myös vierailla säännöllisesti Statsterissa. Tästä huolimatta albumivirta helpottaa kuuntelujen lisäämistä Statsteriin niiden käyttäjien osalta, jotka käyttävät Last.fm:ää.

5.2 Palvelurajapinta

Statster-verkkopalveluun on mahdollista lisätä kuunteluita suoraan käyttäjän koneelta oheishjelmien avulla palvelun tarjoaman palvelurajapinnan ansiosta. Palvelurajapinta sisältää yksinkertaisen Kuuntelun lisäys -rajapinnan, jonka avulla kuunteluita on mahdollista lisätä ohjelmallisesti. Tämä rajapinta on yhdensuuntainen. Rajapinnan avulla ei ole mahdollista hakea tietoa palvelusta. Albumikuuntelutietojen ohjelmalliseen hakemiseen Statster tarjoaa kuunteluvirta-rajapinnan. Tämän rajapinnan avulla viimeisimmät kuuntelut on mahdollista hakea palvelusta joko CSV- (Comma-Separated Values)¹ tai RSS-muodossa (Really Simple Syndication)². Jätämme kuunteluvirta-rajapinnan dokumentoinnin tästä työstä pois, koska se ei liity

¹<http://tools.ietf.org/html/rfc4180>

²<http://www.rss-specifications.com/>

albumikuuntelun lisäämiseen.

Kuuntelun lisäys -rajapinnan toiminta perustuu HTTP:n metodin POST hyödyntämiseen. Metodi POST on suunniteltu tiedon lisäästä silmälläpitäen. Metodi GET on puolestaan tiedon hakemista varten. Tätä metodia käytetään kuunteluvirta-rajapinnassa.

Kuuntelun lisäästä varten rajapinta tarvitsee käyttäjänimen (*username*) ja salasanan (*password*). Salasana annetaan MD5-tiivisteenä. Tunnistautumistiedon lisäksi tarvittavia tietoja ovat artistin (*artist*) ja albumin (*album*) nimi, albumin julkaisuvuosi (*albumyear*) ja kuuntelupäivämäärä. Taulukossa 5.1 on esitetty albumikuuntelun lisäyksessä tarvittavat tiedot.

Taulukko 5.1: Kuuntelun lisäys -rajapinnan pakolliset POST-parametrit, ja niiden esimerkkiarvot.

Parametri	Esimerkki arvo
username	tatu
password	<MD5-tiiviste>
artist	Pink Floyd
album	Wish You Were Here
albumyear	1975
day	07
month	04
year	2011

Kuuntelupäivämäärä on jaettu parametreihin päivä (*day*), kuukausi (*month*) ja vuosi (*year*). Yksinumeroiset päivät ja kuukaudet eivät välttämättä tarvitse nollaa eteen. Albumin julkaisuvuosi (*albumyear*) voidaan jättää nolaksi, jos se ei ole tiedossa, mutta parametriä ei saa jättää tyhjäksi. Lähetettävään kuunteluun on myös mahdollista liittää oheistietoa. Ne on esitetty taulukossa 5.2.

Taulukko 5.2: Kuuntelun lisäys -rajapinnan vapaavalintaiset POST-parametrit, ja esimerkkiarvot.

Parametri	Esimerkki arvo
genre	Progressive Rock, Psychedelic Rock
keyword	David Gilmour, British
format	file
player_name	foobar2000

Kuunteluun liitettävällä oheistiedolla on mahdollista kuvata albumia ja sitä, miten se ollaan kuunneltu. Tyylilaji (*genre*) ja avainsana (*keyword*) kuvaavat albumia. Niiden arvoiksi on mahdollista antaa useampia pilkulla erotettuna. Kuuntelua kuvaavaa tietoa ovat media (*format*), jolta albumi ollaan kuunneltu, ja soitin (*player_name*), jolla albumi ollaan kuunneltu.

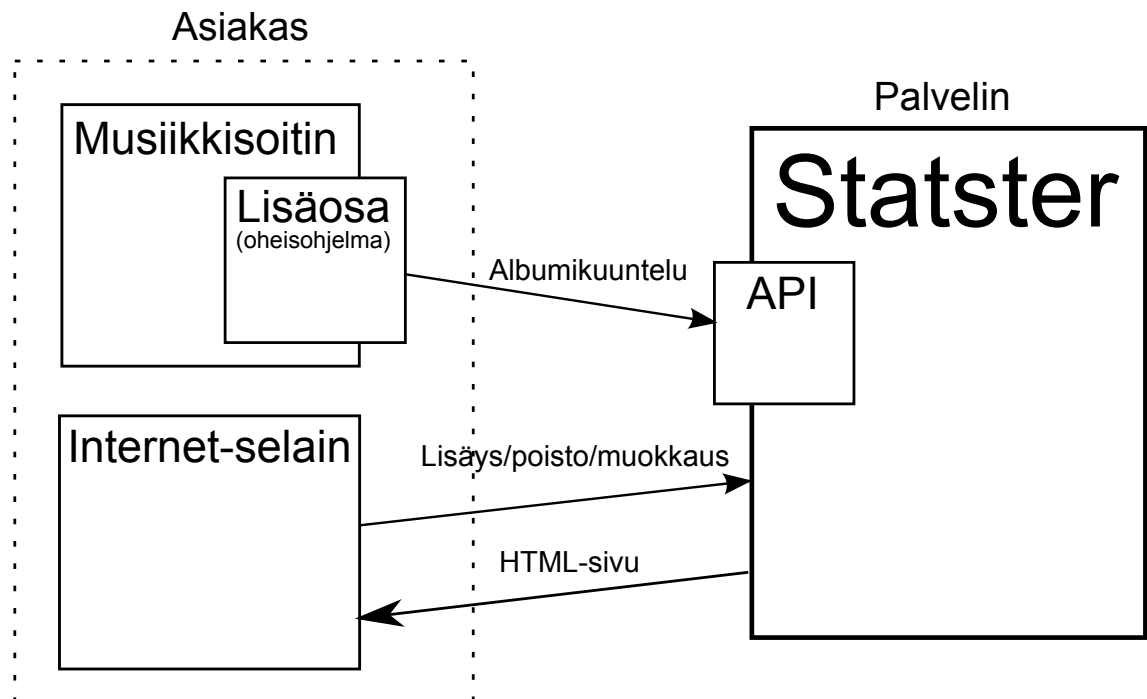
Kuuntelun lisäys -rajapinta vastaa aina takaisin asiakkaalle saatuaan HTTP-pyynnön (request). HTTP-vastaus (response) sisältää tiedon kuuntelun lisäämisen onnistumisesta tai epäonnistumisesta. Tieto on saatavilla joko HTTP:n otsikkoken-
tän status-koodina tai tekstinä vastauksen dokumenttiosassa. Taulukosta 5.3 nähdään, mistä kuuntelun lisäämisen onnistuminen voidaan lukea ja rajoitetusti se, mitä voidaan tietää lisäyksen epäonnistumisesta. Suositeltava tapa tulkita kuuntelun lisäys onnistuneeksi on käyttää HTTP:n otsikon status-koodia.

Taulukko 5.3: Kuuntelun lisäys -rajapinnan HTTP-vastauksen sisältö.

Lisäys	Status-koodi/teksti	Dokumentin sisältö
Onnistui	201/Created	ADDLISTENING_SUCCESS
Epäonnistui	210/Empty tags	EMPTY_TAGS
	220/Incorrect login information	LOGIN_ERROR

5.3 Oheisohjelmat

Statsterin oheisohjelmat ovat sovelluksia, joiden tehtävä on tehdä palvelun käytöstä helpompaa. Ne sijaitsevat käyttäjän koneella, eivät palvelimella, jolla Statster sijaitsee. Oheisohjelmien toiminta perustuu Statsterin tarjoamaan palvelurajapintaan. Alla olevassa kuvassa 5.3 on esitetty oheisohjelman suhde Statster-verkkopalveluun. Kuvassa on myös nähtävillä Internet-selain, jonka avulla palveluun pystytään olemaan vuorovaikutuksessa oheisohjelmaa monipuolisemmin.



Kuva 5.3: Lisäosan suhde soittimeen ja Statsteriin.

Statsteria varten on tehty useita oheisohjelmia, joiden avulla albumikuunteluita on mahdollista lähettää palveluun suoraan musiikkisoittimesta. Olemassa olevat oheisohjelmat ovat musiikkisoittimiin liitettäviä lisäosia. Näiden lisäosien tarkoitus on mahdollistaa kattava kuuntelutietojen kerääminen helposti ilman, että käyttäjän tarvitsee lisätä kuuntelemansa albumit palveluun itse yksitellen. Lisäosat tarjoavat täysin automatisoidun kuuntelun tunnistamisen ja lähettämisen. Käyttäjän tehtäväksi jää ainoastaan kappaleiden sisältökuvauksien oikeellisuudesta huolehtiminen ja tietenkin musiikin kuunteleminen.

5.3.1 Lisäosien toimintaperiaate

Kuunteluntunnistajan toiminta perustuu kappaleiden sisältökuvauksien hyödyntämiseen. Sisältökuvaus on kappaletta kuvaavaa tietoa (metatietoa). Metatieto on yleisesti liitetty musiikkitiedostoihin ID3-muodossa³. Statster vaatii, että kappaleiden sisältökuvauksissa on vähintään artistin ja albumin nimi, myös julkaisuvuosi on palvelun kannalta hyödyllinen. Näiden tietojen pohjalta lisäosa voi toimia ainoastaan manuaalisesti. Käyttäjän täytyy tällöin valita lähetettävät albumi käsin. Automaattinen kuuntelun tunnistus vaatii, että kappaleiden sisältökuvauksissa on artistin ja albumin nimen lisäksi kappaleen numero ja otsikko.

Toteutuksen näkökulmasta kuunteluntunnistajan toiminta perustuu ensinnäkin siihen, että soitinta on mahdollista laajentaa, ja toiseksi soittimelta täytyy saada lisäosalle soivan kuunneltavan albumin kappaleista. Käytännössä tämä tarkoittaa sitä, että soittimen täytyy pitää yllä musiikkikirjastoa, josta kappaleita on mahdollista hakea. Musiikkikirjasto on eräänlainen soittimen toteuttama metatietokanta. Haku tehdään musiikkikirjastoon artistin ja albumin nimen avulla ja myös julkaisuvuodella, jos se on saatavilla. Julkaisuvuotta hyödyntäen saadaan varmemmin oikea albumi. Julkaisuvuoden avulla rajataan pois sellaiset albumit, joilla on sama nimi ja sama artisti, mutta eri julkaisuvuosi. Kappalelistaa käytetään kuuntelun etenemisen seurantaan.

Albumin kappaleista haetaan aina, kun aletaan kuuntelemaan kappaletta numero yksi. Uuden kappaleen alkaessa sen tietoja verrataan albumin kappalelistaan, joka on saatu ensimmäisen kappaleen yhteydessä. Jos kappale vaihtui kappaleeseen, jonka numero on edellisen kappaleen numero kasvatettuna yhdellä, albumia kuunnellaan tällöin järjestyksessä. Albumi voidaan todeta kuunnelluksi ja sen tiedot lähettää Statsteriin, kun albumin kaikki kappaleet on kuunneltu järjestyksessä viimeiseen saakka. Tämä toimintaperiaate perustuu siihen ajatukseen, että Statster tilastoi kokonaisia albumeja, ei yksittäisiä kappaleita.

³<http://www.id3.org>

5.3.2 Olemassa olevat oheisohjelmat

Statsteria varten on tehty automaattinen kuunteluntunnistin kolmelle soittimelle. Käyttäjärjestelmässä Windows toimivalle mediasoittimella Winamp ⁴ ja Linuxissa toimiville musiikkisoittimille Amarok 1.4⁵ ja Exaile 0.3.2⁶. (*Statster – lisäosat*, s.a.)

Amarokille tehty lisäosa on ominaisuuksiltaan monipuolisin. Sen käyttöönotto onnistuu graafisen käyttöliittymän avulla. Tarvittavat tiedot ovat käyttäjätunnus ja salasana. Konfiguroinnin jälkeen lisäosa toimii automaattisesti olettaen, että kuunneltavat kappaleet ovat soittimen ylläpitämässä musiikkikirjastossa. Kun kuunteluntunnistaja havaitsee albumin kaikki kappaleet kuunnelluiksi, se lähettää albumin tiedot Statsteriin. Lähetysten onnistuessa lisäosa ilmoittaa käyttäjälle onnistumisesta ja epäonnistuessa käyttäjälle ilmoitetaan tapahtuneesta virheestä. Verkkovirheen yhteydessä albumikuuntelu tallennetaan välimuistiin (cache) odottamaan verkon toiminnan palautumista. Verkon toiminnan palauduttua normaaliksi välimuistissa olevat albumit lähetetään seuraavan albumikuuntelun yhteydessä. Käyttäjän on mahdollista lähettää albumikuunteluita käsin, jos automaattilähetys ei jostain syystä tunnista albumia kuunnelluiksi. Tämä tapahtuu soittolistan kontekstivalikosta, joka saadaan auki painamalla hiiren kakkospainiketta kuunneltavan albumin kappaleen kohdalla.

Winampille tehty kuunteluntunnistaja sisältää lähestulkoon kaikki samat ominaisuudet mitä Amarokissa on, lukuunottamatta graafista konfigurointia ja ilmoitusviestejä. Winampissa käyttäjätunnus ja salasana täytyy pistää käsin suoraan tekstitiedostoon. Tästä johtuen salasana annetaan suoraan MD5-tiivisteenä. Myös lisäosan toiminnasta kertovat ilmoitusviestit puuttuvat, joten toiminta on täysin huomaamaton.

Exailen kuunteluntunnistajasta puuttuu Amarokkiin verrattuna ilmoitusviestit ja käsin kuuntelun lähettäminen. Yhteisiä ominaisuuksia ovat graafinen käyttöliittymä konfiguroimiseen, kuuntelun automaattilähetys ja välimuisti verkkovirheiden varalta.

⁴<http://www.winamp.com/>

⁵<http://amarok.kde.org/>

⁶<http://www.exaile.org/>

6. PROTOTYYPITOTEUTUS

Diplomityön teknisenä kontribuutiona rakennettu prototyyppitoteutus¹ testaa ulkoisen, verkossa sijaitsevan musiikin metatietokannan mahdollisuuksia albumikuuntelun tunnistamiseen. Metatietokannaksi on valittu MusicBrainz. Soitinriippumattoman kuuntelun tunnistamisen ensimmäistä vaihetta eli rajapintoja, joiden avulla soivan kappaleen tiedot saadaan soittimilta, ei testata.

Prototyyppitoteutuksen alustana käytetään foobar2000-musiikkisoitinta². Soittimeen on saatavilla komponentti nimeltä WSH Panel Mod³, jonka avulla komentosarjojen suorittaminen on mahdollista.

Tässä luvussa perehdytään prototyypin toteutusympäristöön eli foobar2000-soittimeen ja WSH Panel Mod -komponenttiin sekä itse toteutukseen. Toteutusta tarkastellaan sen tärkeimpien ominaisuuksien osalta, joiden avulla albumikuuntelun tunnistaminen on mahdollista käyttäen vain artistin ja kappaleen nimeä.

6.1 Toteutusympäristö

Prototyypin toteutusympäristö koostuu käyttöjärjestelmästä Windows 7, johon on asennettu foobar2000-soittimen versio 1.1.6. Prototyyppitoteutus on tehty osaksi kyseistä soitinta käyttämällä apuna komponenttia WSH Panel Mod. Yksi suuri etu tässä toteutustavassa on testauksen suorittamisen helppous. Komponentin ja soittimen välillä on valmis yhteys, jota hyödyntäen soittimen koko musiikkikirjasto on prototyypin käytettävissä.

Foobar2000-soitin on suunniteltu toimimaan ainoastaan Windows-käyttöjärjestelmissä, mutta sitä on myös mahdollista käyttää erillisen ohjelman avulla esimerkiksi Linuxissa. Soitinta jaellaan ilmaisohjelmana. Lähdekoodia ei ole saatavilla. Suljetusta lähdekoodista huolimatta soitin on hyvin mukautettavissa käyttäjien tarpeisiin sen monipuolisen ohjelmointirajapinnan ansiosta. Soittimeen on esimerkiksi tehty laajennoksena vaihtoehtoinen käyttöliittymä nimeltä Columns UI. Soittimen verkkosivuilta on mahdollista ladata SDK (Software Development Kit), jonka avulla lisäosien eli komponenttien kirjoittaminen on mahdollista. (*foobar2000*, 2011)

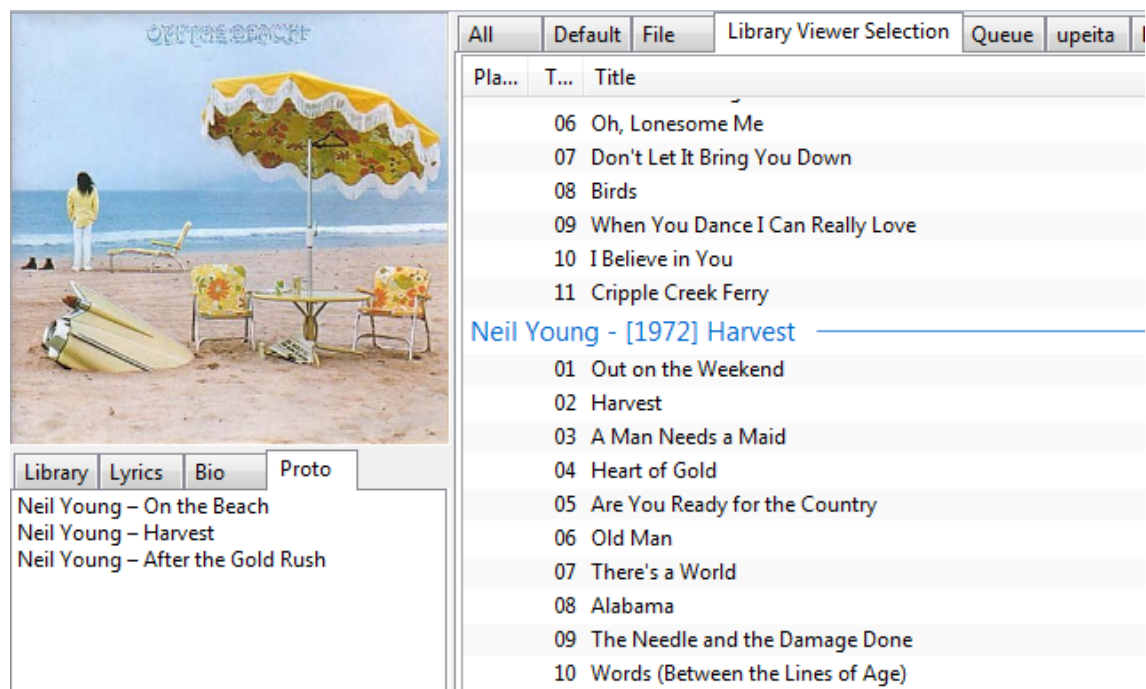
¹<http://statster.info/plugins/AlbumListeningRecognizer.txt>

²<http://www.foobar2000.org/>

³<http://code.google.com/p/foo-wsh-panel-mod/>

6.1.1 WSH Panel Mod -komponentti

Monipuolisen ohjelmointirajapinnan ansiosta albumikuuntelun tunnistajan prototyyppi on myös mahdollista toteuttaa osaksi foobar2000-soitinta komentosarjan muodossa. Apuna käytetään Wangin luomaa WSH Panel Mod -komponenttia (Wang, 2009), josta on käytössä versio 1.4.2. Komponentin avulla soittimen toimintaa on mahdollista laajentaa joko komentosarjan JScript tai VBScript avulla. Komponentti sisältää Scintilla-lähdekoodieditorin⁴, jonka avulla ladattuja komentosarjoja on mahdollista muokata suoraan soittimesta. Komentosarja otetaan käyttöön luomalla uusi käyttöliittymäelementti, johon lisätään WSH Panel Mod -komponentti. Jokainen komentosarja vaatii oman käyttöliittymäelementin, koska sen kautta siihen ollaan vuorovaikutuksessa. Komponentin käyttöliittymä on nähtävillä kuvassa 6.1 vasemmalla alhaalla ”Proto”-nimisessä välilehdessä. Välilehti on eräs foobar2000-soittimen käyttöliittymäelementeistä.



Kuva 6.1: Prototyypitoteutuksen tulostetta WSH Panel mod -komponentin käyttöliittymäelementissä ”Proto”.

WSH Panel Mod -komponenttiin ei ole sisäisesti rakennettu komentosarjoja tulkaavaa moottoria. Komponentti hyödyntää hyödyntää käyttöjärjestelmässä olevaa Windows Scripting Host -ympäristöä (*What Is WSH?*, 2009). Prototyypin toteutusympäristössä käytössä on WSH:n versio 5.8, joka on samalla myös JScript- ja VBScript-kielten versio. (Guy, 2010)

⁴www.scintilla.org/

6.1.2 Käytetyt WSH Panel Mod -komponentin rajapinnat

Albumikuuntelun tunnistajan prototyypin toiminta perustuu WSH Panel Mod -komponentin tarjoamiin rajapintoihin ja callback-funktioihin, joiden avulla soittimen kanssa voidaan olla vuorovaikutuksessa. Callback-funktioista tärkein on *on_playback_new_track*, jota kutsutaan aina kappaleen vaihtuessa. Funktion parametrina saadaan *IFbMetadbHandle* tyyppinen muuttuja nimeltä *metadb*, jonka avulla päästään käsiksi kappaleen metatietoihin.

Komponentin toteuttamista rajapinnoista tärkeimpiä ovat *IFbUtil* ja *IFbTitleFormat*, joiden avulla saadaan muun muassa soivan kappaleen nimi ja artisti. Rajapinnan *IFbUtil* tärkeimmät metodit ovat *TitleFormat* ja *trace*. Metodilla *trace* saadaan tulostettua tekstiä soittimen konsoliin. Metodin *TitleFormat* avulla kappaleiden metatiedoista saadaan haettua tietoa käyttämällä Title formatting -tyyppisiä lausekkeita (*Foobar2000:Title Formatting Reference*, 2010). Se palauttaa *IFbTitleFormat*-tyyppisen objektin, jonka tarjoamalla metodilla *EvalWithMetadb* kappaleen metatiedot saadaan haettua antamalla kyseiseen kappaleeseen viittaava objekti *metadb*, joka on tyyppiä *IFbMetadbHandle*. Objekti *metadb* saatiin callback-funktion *on_playback_new_track* parametrina. Artistin nimi saadaan haettua soittimelta edellä kuvattujen rajapintojen avulla seuraavasti:

```
fb.TitleFormat("%artist%").EvalWithMetadb(metadb)
```

Komponentti tarjoaa globaalin muuttujan nimeltä *fb*, jonka avulla rajapintaa *IFbUtil* voidaan käyttää komentosarjassa. Muita globaaleja muuttujia ovat *window*, *gdi* ja *utils*. Muuttuja *window* toteuttaa rajapinnan *IFbWindow*, jonka avulla komentosarjan käyttöliittymäelementin ikkunaan pystytään käsittelemään; *gdi* toteuttaa rajapinnan *IGdiUtils*, jonka avulla grafiikkaa pystytään luomaan ikkunaan; ja *utils* toteuttaa rajapinnan *IWSHUtils*, joka tarjoaa joukon komponentin toteuttamia apu-metodeja.

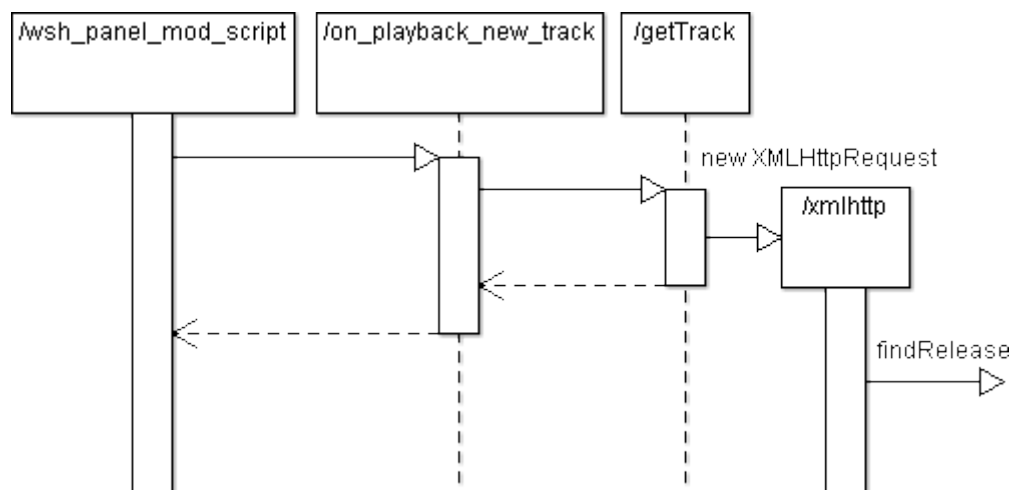
6.2 Toteutus

Prototyypin toteutuksessa on käytetty apuna komponentille WSH Panel Mod tehtyä komentosarjaa nimeltä Playcount sync, jonka avulla Last.fm-palvelun kuuntelutilastot voidaan tuoda näkyville foobar2000-soittimeen (*HOW TO: synchronise your last.fm playback stats with foobar*, s.a.). Playcount sync on kirjoitettu JScript-kieltä käyttäen, tästä syystä oli luontevaa valita albumikuuntelun tunnistajan prototyypin kieleksi myös JScript.

6.2.1 Yleiskuvaus

Prototyypin komentosarjan suoritus alkaa WSH Panel Mod -komponentin kutsuessa callback-funktiota *on_playback_new_track*. Funktion suorituksen aikana tehdään kaikki välttämättömät asiat, kuten kappaleen metatietojen hakeminen soittimelta ja kappalehaun (*recording*) HTTP-pyyntön lähettäminen MusicBrainzin palvelurajapinnalle. Callback-funktioon ei jäädä odottamaan palvelurajapinnan palauttamaa vastausta, koska foobar2000-soittimen soittolista jumittuu callback-funktiokutsujen ajaksi. Soittimen käytön sujuvuuden kannalta on tärkeää saada suoritus mahdollisimman nopeasti pois funktiosta *on_playback_new_track*.

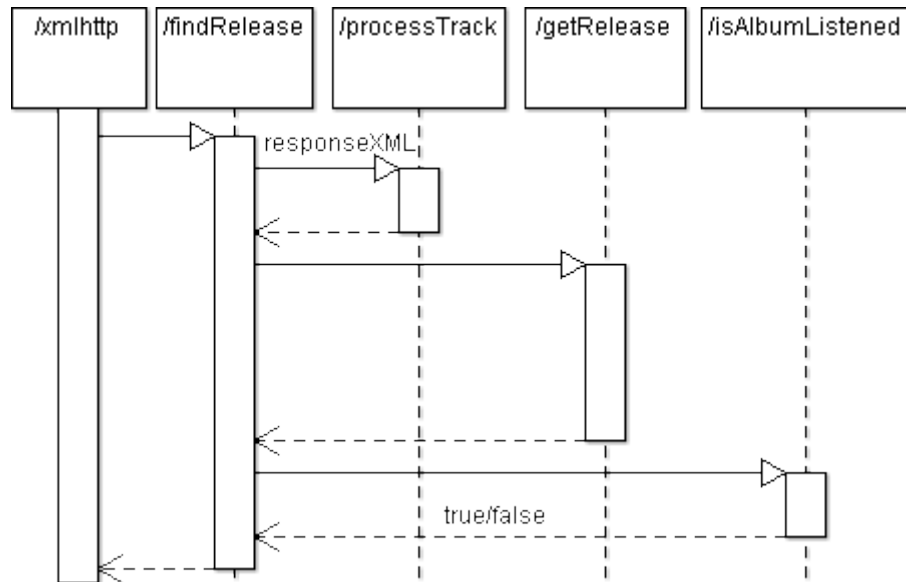
Komentosarjan suoritus saadaan irtautumaan callback-funktiosta käyttämällä *XMLHttpRequest*-objektia asynkronisesti. Objektin *XMLHttpRequest* avulla tehdään esimerkiksi HTTP:n GET ja POST pyynnot JavaScriptin kaltaisissa komentosarjakielissä, kuten JScript (Kesteren, 2010). HTTP-pyyntöjen käsittely saadaan asynkroniseksi kutsumalla metodia *open* siten, että sen kolmannelle parametrille annetaan arvo *true*. Myös objektin tapahtumakuuntelijan *onreadystatechange* arvoksi tulee asettaa funktio, jossa vastauksen käsittely tehdään, kun vastaus on kokonaisuudessa saatu MusicBrainzilta. Prototyypin tapauksessa tapahtumankuuntelijan funktiossa tullaan suorittamaan kaikki loput albumikuuntelun tunnistuksen vaatimat asiat. Kuvassa 6.2 on esitetty suorituksen kulku callback-funktion kutsusta sieltä poistumiseen saakka.



Kuva 6.2: Ensimmäinen osa prototyypin komentosarjan (*wsh_panel_mod_script*) toiminnan vaiheista sekvenssikaavion muodossa.

Callback-funktiossa *on_playback_new_track* kutsutaan funktiota *getTrack*, jossa luodaan *XMLHttpRequest*-tyyppinen objekti nimeltä *xmlhttp*. Objektille tehdään tarpeelliset alustustoimenpiteet, jonka jälkeen HTTP:n GET-pyyntö lähetetään. Pyyntön lähettämisen jälkeen suoritus palaa callback-funktioon ja sieltä suoraan

WSH Panel Mod -komponentille. Objekti *xmlhttp* jatkaa olemassaoloaan funktion *on_playback_new_track* suorituksen loppumisen jälkeen. Objektin *xmlhttp* tapahtumankuuntelijan *onreadystatechange* arvoksi asetettiin funktio *findRelease*, jota kutsutaan kun GET-pyynnön vastaus on latautunut kokonaan. Funktiossa *findRelease* suoritetaan kaikki loput albumikuuntelun tunnistamisen vaatimat toimenpiteet, jotka on esitetty kuvan 6.3 sekvenssikaaviossa.



Kuva 6.3: Viimeinen osa prototyypin komentosarjan (*wsh_panel_mod_script*) toiminnan vaiheista sekvenssikaavion muodossa.

MusicBrainziin lähetetyn *recording*-haun vastauksen latauduttua suoritus siirtyy funktioon *findRelease*. Funktio hoitaa vastauksena saadun XML-dokumentin käsittelyn kutsumalla funktiota *processTrack*, jonka avulla saadaan luettua kappaleen julkaisut. Tämän jälkeen haetaan yhdelle julkaisulle kappaleista funktiolla *getRelease*. Julkaisun valintaa käsitellään alakohdassa 6.2.3. Funktiossa tehdään synkroninen *release*-haku, jolla albumin kappaleista saadaan MusicBrainzista. Myös saadun vastauksen käsittely tehdään kyseisessä funktiossa. Kappalelistojen avulla albumin kuuntelemisen etenemistä voidaan seurata. Kuuntelun etenemisen seuranta tehdään funktiossa *isAlbumListened*, joka palauttaa totuusarvon *true* tai *false*. Totuus arvo *true* palautetaan, kun albumi ollaan kuunneltu.

6.2.2 Julkaisujen hakeminen kuunneltavalle kappaleelle

Albumikuuntelun tunnistaja tarvitsee kahdenlaista tietoa kuuntelun tunnistamiseen: julkaisut, joilla kuunneltava kappale esiintyy ja kuunneltavien kappaleiden julkaisujen kappalelistat. Kappaleen julkaisut saadaan haettua yhdellä HTTP:n GET-pyyntöillä käyttäen MusicBrainzin hakupalvelimen indeksoitua *recording*-hakua. Saa-

duista julkaisuista valitaan vain yksi, jonka kappaleista haetaan *MBID*-tunnuksella käyttäen palvelurajapinnan suoraa *release*-hakua. Julkaisuksi pyritään valitsemaan se, joka todennäköisimmin on kuunneltava albumi. Apuna julkaisun valinnassa käytetään kuunteluhistoriaa ja sitä, miten julkaisut sijoittuvat *recording*-haun tuloksista. Julkaisun valinnan tekevää osaa albumikuuntelun tunnistajassa kutsutaan kuuntelupuskuriksi. Kuuntelupuskuriin perehtymistä ennen selvitetään tiedot, joiden pohjalta se toimii.

Tiedot, joiden avulla kuuntelupuskuri toimii saadaan MusicBrainzin *recording*-haulla. Soittimelta saaduilla artistin ja kappaleen nimellä haetaan julkaisuja kolmella eri hakutavalla synkronisessa järjestyksessä. Kun tuloksia saadaan, jäljelle jääneitä hakuja ei enää suoriteta. Ensimmäinen *recording*-haun kysely rakennetaan artistin ja kappaleen nimestä listauksen 6.1 mukaisesti. Kuvan 6.2 sekvenssikaaviossa *recording*-haun kyselyistä huolehtii funktio *getTrack*.

Listaus 6.1: Haun *recording* ensimmäinen kysely.

```
query=smells like teen spirit* AND artist:"nirvana"
```

Kappaleen nimen lopussa olevalla jokerimerkillä (*) saadaan parempia tuloksia, jos nimi on jostain syystä katkennut, esimerkiksi jos käytössä on ID3v1-tagit. ID3-tagin versio 1 sallii maksimissaan 30 merkkiä kappaleen otsikkokenttään ja artistin nimikenttään (*What is ID3 (v1)?*, s.a.). Jokerimerkin käyttöön liittyy myös ongelmia. Esimerkiksi jos se laitetaan ?-merkin perään, ei saada tuloksia lainkaan. Myös ASCII-merkkeihin kuulumattomilla, kuten merkeillä ä, ö, å, on sama vaikutus.

Haun tulokset voivat sisältää useita kappaleita ja kappaleet voivat sisältää useita julkaisuja. Esimerkiksi Nirvanan kappaleelle *Smells Like Teen Spirit* saadaan 108 erinimistä julkaisua, kun parametrin *limit* arvo on 100. Julkaisujen määrä voi olla suurempi kuin *limit*-arvo, koska tietty kappale voi olla useammalla julkaisulla. On myös mahdollista, että ensimmäisellä *recording*-haulla ei saada ollenkaan tuloksia. Tämän jälkeen tuloksia yritetään saada kolmella tavalla:

1. poistamalla kappaleen nimen lopusta ylimääräinen,
2. hakemalla pelkällä kappaleen nimellä ja
3. hakemalla pelkällä artistin nimellä.

Ylimääräisen poistamisella tarkoitetaan sellaisen tekstin poistamista, mikä on kaari- tai hakasulkeitten sisällä, tai jos kappaleen nimessä on tavuviivalla (-) erotettuna tekstiä. Kaikki kolme *recording*-hakua tehdään funktiossa *findRelease*, joka on nähtävillä kuvassa 6.3. Hakuja ei ole kuvaan merkitty, ne tehdään funktion *getTrack* avulla.

Haun palauttamasta XML-dokumentista luetaan talteen kaikki julkaisut, joilla kappale esiintyy. Tiedot luetaan ja tallennetaan funktiossa *processTrack*, joka on nähtävillä edellisen kohdan kuvassa 6.3. Dokumentin sisältämä tieto on ryhmitelty *recording*-elementteittäin relevanttiuden perusteella laskevaan järjestykseen. Elementit *recording* sisältävät joukon *release*-elementtejä. Nämä elementit sisältävät tiedot julkaisuista, joilla kappale esiintyy. Jokaisen julkaisun tiedot tallennetaan *Track*-tyyppiseen objektiin ja tämä objekti tallennetaan assosiatiiviseen taulukkoon (tyyppiä *Object*), joka sisältää tietyn julkaisun vain kerran. Julkaisut yksilöidään niiden *MBID*-tunnuksen avulla, joka toimii myös taulukossa avaimena. Objektin *track* sisältämät attribuutit on esitetty taulukossa 6.1.

Taulukko 6.1: Objektin *Track* attribuutit.

Attribuutti	Kuvaus
title	Kappaleen nimi.
artist_name	Kappaleen tehneen artistin nimi.
release_title	Julkaisun nimi, jolla kappale on.
mbid	Tunniste, joka yksilöi julkaisun.
type	Voi saada jonkin arvoista: <i>album, single, EP, compilation, live, spokenwords, soundtrack, interview, audiobook, remix, other</i> .
status	Voi saada jonkin arvoista: <i>official, promotion, bootleg, pseudo-release</i> .
format	Median tyyppi, jolla kappale on julkaisu. Voi olla esimerkiksi <i>CD, DVD, Vinyl</i> .
number	Kappaleen järjestysnumero medialla.
count	Sen median sisältämien kappaleiden määrä, jolla kyseinen kappale on.
total_count	Julkaisun kaikkien medioiden sisältämien kappaleiden yhteismäärä.
disc_number	Julkaisun sisältämien medioiden määrä.
date	Päivämäärä, jolloin kappale on julkaistu. Voi sisältää joko pelkän vuoden tai tarkemman päivämäärän muodossa <i>pp-kk-vvvv</i> .

Funktioista *processTrack* voidaan palata takaisin funktioon *findRelease*, kun kaikkien kappaleiden tiedot ollaan saatu luettua *Track*-objekteihin ja lisättyä taulukkoon. Paluuarvona funktiolta saadaan luotu *Object*-taulukko, joka lisätään kuuntelupuskuriin funktiossa *findRelease*.

6.2.3 Oikean julkaisun valinta

Albumin kuuntelemisen etenemisen seurantaan tarvitaan julkaisu, jota käytetään vertailukohtana. Tästä syystä julkaisun tulee olla mahdollisimman suurella todennäköisyydellä kuunneltavan albumin julkaisu. Vertailujulkaisun valinnan tekee kuun-

telupuskuriksi kutsuttu komponentti, joka pitää yllä tietyn mittaista kuunteluhistoriaa. Kuuntelupuskuri on tyyppiä *Object*, jonka nimi on *TrackBuffer*. Siitä luotu instanssi on globaali objekti, joka luodaan komentosarjan käynnistymisen yhteydessä. Kuuntelupuskuria ei ole merkitty kuvien 6.2 ja 6.3 sekvenssikaavioihin tilan puutteen vuoksi. Kuvan 6.2 sekvenssikaaviossa objektin *TrackBuffer* elinkaari on sidottu komentosarjan *wsh_panel_mod_script* elinkaareen.

Kuuntelupuskuri taltioi tietyn määrän funktiolta *processTrack* saatuja *Object*-tyyppisiä taulukoita, joiden alkioina on *Track*-objekteja. Object-taulukko lisätään kuuntelupuskurin sisäiseen *Array*-tauluun, jonka koko on kiinteä. Kuuntelupuskurin koko määritetään luomisvaiheessa. Kolmen mittaisella kuuntelupuskurilla saadaan jo hyviä tuloksia. Sen sisältönä on tällöin kolmen viimeksi kuunnellun kappaleen julkaisutiedot. Taulukossa 6.2 on nähtävillä kuuntelupuskurin sisältö kuvitteellisessa tilanteessa, jossa kuunnellaan artistin *Bob Dylan* albumia *Blood on the Tracks*.

Taulukko 6.2: Kolmen alkion pituisen *TrackBuffer*-objekti sisältö.

	1	2	3
1	Away We Go	Blood on the Tracks	John Wesley Harding
2	Dylan	John Wesley Harding	Blood on the Tracks
3	Blood on the Tracks	Dylan	Dylan
	Tangled Up In Blue	Simple Twist Of Fate	You're A Big Girl Now

Taulukon numeroidut sarakkeet kuvaavat kuuntelupuskurin tallennuspaikkoja, jotka sisältävät kuunneltujen kappaleiden avulla saatuja julkaisuja. Kappaleen nimet ovat näkyvillä viimeisellä rivillä. Ensimmäinen sarake sisältää julkaisut, jotka on haettu pisimpään kuuntelupuskurissa olleella kappaleella. Keskimmäinen, sarake 2, sisältää kuunneltavaa kappaletta edeltävällä kappaleella saadut julkaisut. Sarake 3 sisältää puolestaan kuunneltavan kappaleen tiedoilla haetut julkaisut. Taulukon numeroidut rivit osoittavat julkaisun sijoituksen hakutuloksissa eli mitä pienemmällä rivinumerolla julkaisu on sitä korkeammalla se on hakutuloksissa. Rivin 1 julkaisut ovat olleet hakutuloksissa ensimmäisinä ja niin edelleen.

Julkaisu, joka valitaan albumikuuntelun etenemisen vertailukohteeksi, haetaan puskurista laskemalla ensin julkaisujen esiintymismäärät yhteen. Taulukon 6.2 tapauksessa julkaisuja *Dylan* ja *Blood on the Tracks* on kolme, julkaisua *John Wesley Harding* on kaksi ja *Away We Go* -julkaisua on vain yksi. Tämän jälkeen suosituimpien julkaisujen joukosta valitaan se, joka esiintyy useimmiten hakutulosten kärjessä. Julkaisu *Blood on the Tracks* on kahdesti korkeammalla sijoituksella kuin julkaisu *Dylan*, joten *Blood on the Tracks* valitaan kuuntelun etenemisessä käytettäväksi vertailujulkaisuksi. Alla olevassa listassa on esitetty kuuntelupuskurin sisältö, kun julkaisut ollaan järjestetty edellä mainittujen kriteerien mukaisesti:

1. 3x Blood on the Tracks (album)
2. 3x Dylan (compilation)
3. 2x John Wesley Harding (album)
4. 1x Away We Go (soundtrack)

Mitä suurempi listan järjestysnumero on sitä epätodennäköisemmin julkaisu on albumi, jota kuunnellaan. Nyt voidaan siirtyä julkaisun kappalelistan hakemiseen, kun tiedossa on julkaisu, jota todennäköisesti kuunnellaan.

6.2.4 Kappalelistan hakeminen julkaisulle

Julkaisun kappalelista saadaan haettua suoralla *release*-haulla käyttäen *MBID*-tunnusta, joka saadaan kuuntelupuskurin palauttamasta *Track*-objektista. Haku ja sen käsittely tehdään funktiossa *getRelease* (kuva 6.3). Funktio ottaa parametriksi julkaisun *MBID*-tunnuksen. Tämän jälkeen funktiossa muodostetaan *MBID*-tunnuksen avulla *release*-haun pyyntö, johon myös liitetään *inc*-parametri arvolla "*recordings+artists+release-groups*". Parametrin arvolla *recordings* ilmaistaan, että julkaisun tietoihin halutaan sisällyttää kappalelista, arvolla *artists* saadaan julkaisun tehneen artistin nimi ja arvolla *release-groups* saadaan julkaisun tyyppi.

Haun tuloksena saadun XML-dokumentin tiedot tallennetaan objektin *Release* attribuutteihin, jotka ovat nähtävillä taulukossa 6.3. Julkaisun nimi saadaan *release*-elementin ensimmäisen lapsielementin *title* arvosta ja julkaisun status saadaan seuraavasta lapsielementistä *status*. Muita suoraan elementin *release* lapsena olevia elementtejä, jotka tallennetaan objektiin *Release*, ovat *date* ja *release-group*. Julkaisun tyyppi saadaan elementin *release-group* attribuutin arvosta *type*. Artistin nimi saadaan *release* elementin lapsen *artist-credit* sisällä olevasta ensimmäisestä *artist*-elementistä.

Taulukko 6.3: Objektin *Release* attribuutit.

Attribuutti	Kuvaus
<i>title</i>	Julkaisun nimi.
<i>artist</i>	Julkaisun tehneen artistin nimi.
<i>date</i>	Julkaisupäivämäärä.
<i>discs</i>	Tyyppiä <i>Array</i> , joka sisältää <i>Disc</i> -objekteja.
<i>type</i>	Voi olla jokin arvoista: <i>album</i> , <i>single</i> , <i>EP</i> , <i>compilation</i> , <i>live</i> , <i>spokenwords</i> , <i>soundtrack</i> , <i>interview</i> , <i>audiobook</i> , <i>remix</i> , <i>other</i> .
<i>status</i>	Voi olla jokin arvoista: <i>official</i> , <i>promotion</i> , <i>bootleg</i> , <i>pseudo-release</i> .
<i>length</i>	Julkaisun medioiden lukumäärä.
<i>mbid</i>	Yksilöllinen tunniste.

Julkaisun sisältämät kappalelistat tallennetaan *Release*-objektin attribuuttiin *discs*. Jokaista julkaisun mediaa varten luodaan *Disc*-tyyppinen objekti, jonka attribuutit ovat esitetty taulukossa 6.4.

Taulukko 6.4: Objektin *Disc* attribuutit.

Attribuutti	Kuvaus
<i>title</i>	Median otsikko, jos sellainen on saatavilla.
<i>format</i>	Median tyyppi. Voi olla esimerkiksi <i>CD</i> , <i>DVD</i> , <i>Vinyl</i> .
<i>tracks</i>	Tyyppiä <i>Array</i> , joka sisältää kappaleiden nimet <i>String</i> -merkkijonoina.
<i>length</i>	Kappaleiden lukumäärä.

Tiedot julkaisun sisältämistä kappaleista on koottuna medioittain *release*-elementin lapsen *medium-list* sisälle, joka puolestaan koostuu *medium*-elementeistä. Elementti *medium* sisältää *position*- ja *track-list*-elementit, ja voi sisältää myös *title*-elementin. Elementin *position* arvona on median järjestysnumero. Median numeroa ei tarvitse lukea, koska *medium*-elementtien järjestys kertoo myös sen. Elementistä *track-list* saadaan luettua median sisältämien kappaleiden nimet *Array*-tyyppiseen objektiin, joka asetetaan objektin *Disc* attribuutin *tracks* arvoksi.

6.2.5 Albumikuuntelun tunnistaminen

Viimeinen vaihe komentosarjan suorituksessa on selvittää kuunneltavan kappaleen sijainti vertailujulkaisulla ja lähettää julkaisun tiedot Statsteriin kun albumi ollaan kuunneltu loppuun. Prototyypin tapauksessa albumin loppuun kuunteleminen tarkoittaa kahden kappaleen kuuntelemista peräkkäin. Nämä kappaleet ovat toiseksi viimeinen ja sitä edeltävä.

Albumikuuntelun tunnistus tehdään kahden kappaleen perusteella, koska pidemmän kuunteluhistorian käyttäminen heikentää kuuntelun tunnistamista. Kahdelle kappaleelle saadaan huomattavasti suuremmalla todennäköisyydellä oikea julkaisu kuin kaikille albumin kappaleille. Ideaalisinta Statsterin kannalta olisi kuitenkin tehdä albumikuuntelun tunnistus albumin kaikkien kappaleiden perusteella. Tällöin albumi oltaisiin kuunneltu vasta silloin, kun kaikki kappaleet ollaan kuunneltu järjestyksessä ensimmäisestä viimeiseen. Tämä ei kuitenkaan ole vielä mahdollista prototyypissä, koska jokaiselle albumin kappaleella ei välttämättä saada haettua oikeaa julkaisua. Puutteet oikean julkaisun hakemisessa johtuvat yleensä kuunneltavan kappaleen kirjoitusasun eroavaisuuksista verrattuna MusicBrainzissa oleviin.

Se, miksi viimeinen kuuntelun tunnistamisessa käytettävä kappale on albumin toiseksi viimeinen, johtuu mahdollisista eroavaisuuksista kuunneltavassa albumissa tai vertailujulkaisussa. Albumi pystytään tunnistamaan, jos esimerkiksi viimeinen

kappale puuttuu albumilta. Albumi pystytään myös tunnistamaan, jos vertailujulkaisuksi tulee julkaisu, jossa on yksi kappale enemmän kuin kuunneltavassa julkaisussa.

Tarvittavat tiedot albumikuuntelun tunnistamisessa ovat kuunneltavan sekä sitä edeltävän kappaleen nimi sekä vertailujulkaisun kappalelista. Kuunneltava ja sitä edeltävä kappale saadaan kuuntelupuskurista. Tästä johtuen prototyyppi ei pysty tunnistamaan albumeita, joissa on vain yksi kappale.

7. ARVIOINTI

Tässä luvussa arvioidaan prototyypitoteutusta ja sen kykyä tunnistaa kuunneltavia albumeita. Soittimien tarjoamat rajapinnat jätetään arvioinnin ulkopuolelle, koska niitä ei toteutettu prototyyppiin.

Ensimmäisessä kohdassa arvioidaan prototyyppiin toteutettuja toimintoja. Toimintojen arvioinnin jälkeen perehdytään suurimpiin puutteisiin, joita on ilmennyt toteutuksen aikana. Puutteet on tärkeää tiedostaa, jotta ne korjattaisiin. Lopuksi esitetään muutamia jatkokehitysideoita liittyen sekä prototyypisovellukseen että Statster-verkkopalveluun.

7.1 Toteutettujen toimintojen arviointi

Toteutettujen toimintojen arviointi noudattaa edellisen luvun kohdassa *Toteutus* (6.2) esitettyjä albumikuuntelun tunnistamisen vaiheita. Arvioinnissa ei perehdytä teknisiin yksityiskohtiin, jotka ovat mahdollistaneet prototyypisovelluksen toteuttamisen.

Albumikuuntelun tunnistus voidaan jakaa osiin prototyypitoteutuksen toiminnan hahmottamisen helpottamiseksi. Ensimmäiseksi haetaan julkaisut, joilla kuunneltava kappale esiintyy. Tämän jälkeen saatavilla olevista julkaisuista päätellään kuunneltavan albumin julkaisu. Lopuksi julkaisulle haetaan kappaleista, jota hyödynnetään albumikuuntelun tunnistamisessa.

7.1.1 Julkaisujen hakeminen kuunneltavalle kappaleelle

Kyselyn muodostaminen. MusicBrainzin *recording*-haku tarjoaa monipuolisen kyselykielen tiedon hakemiseen. Tästä johtuen kyselyn muodostamisella on suuri vaikutus saataviin tuloksiin. Kyselyssä käytettävä jokerimerkki on yksi tällainen asia, joka vaikuttaa saataviin tuloksiin joko kielteisesti tai myönteisesti. Kielteinen vaikutus ilmenee, kun jokerimerkki laitetaan aina kappaleen nimen loppuun, jolloin ei välttämättä saada tuloksia. Jokerimerkkiä ennen oleva sana ei saa sisältää joitain tiettyjä merkkejä tai loppua niihin. Tämä ongelma voidaan kiertää käyttämällä haun kyselyssä sekä alkuperäistä että jokerimerkillistä kappaleen nimeä OR-operaattorilla toisistaan erotettuna ja ympäröitynä sulkeilla. Jos esimerkiksi kuunneltava kappale on Bob Dylanin *Lily, Rosemary And The Jack of Hearts*, josta puuttuu “f Hearts”

lopusta. Tällaisissa tapauksissa jokerimerkin avulla saadaan katkennut sana täsmäämään siihen, mikä sen kuuluu olla eli *of*. Jos kysely tehtäisiin pelkästään katkenneella kappaleen nimellä, ei saataisi albumia, jolla kappale on julkaistu.

Edellisen luvun alakohdassa 6.2.2 esitetyssä listauksessa 6.1 jokerimerkki on laitettu suoraan kappaleen nimen perään, ja kyselyllä saadaan 1661 kappaletta. Hakutulosten joukossa on myös oikea julkaisu. Kyselyä ei kuitenkaan voida muodostaa niin yksinkertaisesti, koska joillain kappaleilla saatetaan saada virheellisiä tuloksia tai ei tuloksia lainkaan. Listauksessa 7.1 on nähtävillä, miten kysely muodostetaan OR-operaattorin ja sulkeiden avulla siten, että jokerimerkkiä voidaan aina käyttää hakukyselyssä.

Listaus 7.1: Alkuperäinen ja jokerimerkillinen kappaleen nimi *recording*-haun kyselyssä.

```
query=(lily, rosemary and the jack o OR
      lily, rosemary and the jack o*) AND
      artist:"bob dylan"
```

Yllä olevan listauksen mukaisella kyselyllä saadaan oikea julkaisu, ja se toimii myös erikoistapauksissa. Kyselyn tulosjoukko sisältää yhteensä 2786 kappaletta. Tämä on huomattavasti enemmän mitä yhden *recording*-haun palauttama vastaus voi sisältää, joka on maksimissaan 100 kappaletta. Listauksen 7.1 kyselyyn liittyy myös yksi erikoinen piirre. Poistettaessa kyselystä kaikki OR-operaattorista lähtien sulkeeseen asti, saadaan tuloksia 1480 kappaletta, joiden joukossa on myös oikea julkaisu. Kappale on tällöin kolmantena tuloksissa, kuten se on myös listauksen 7.1 kyselyn tuloksissa.

Kappalehaun *recording* HTTP-pyyntö tehdään maksimissaan neljä kertaa jos edellinen kysely ei tuota koskaan tuloksia. Ensimmäisen kyselyn epäonnistuttua seuraava tehdään karsimalla ylimääräinen pois. Vaikuttaisi siltä, että ensimmäinen ja toinen kysely on mahdollista yhdistää samaksi kyselyksi käyttäen apuna kyselykielen tarjoamaa OR-operaattoria ja sulkeita. Tällöin saadaan kattavammin tuloksia yhdellä kertaa, ja vältetään yhdeltä ylimääräiseltä HTTP-pyyntöltä. Esimerkiksi jos kuunneltava kappale on *London Calling (Live)* artistilta *The Clash*, kysely on tällöin listauksen 7.2 mukainen.

Listaus 7.2: Yhdistetyn *recording*-haun kysely.

```
query=(london calling \ (live\ ) OR
      london calling \ (live\ ) * OR
      london calling ) AND
      artist:"the clash"
```

Yllä oleva kysely eroaa listauksen 7.1 kyselystä sisältämällä karsitun kappaleen nimen *London Calling*, josta on poistettu lopusta (*live*). Kyselyssä olevat kenoviivat

sanaa *live* ympäröivien sulkeiden edessä kumoavat niiden erikoismerkityksen kyselykielessä. Yhdistetyllä kyselyllä saadaan 77 kappaletta. Haettaessa listauksessa 7.3 esitetyllä tavalla ainoastaan alkuperäisen kappaleen nimen avulla, saadaan kappaleita vain 12. Näistä kappaleista mikään ei ole kuunneltavalta albumilta. Jokerimerkki jätettiin pois, koska sulun perään laitettuna se aiheuttaa tyhjän tulosjoukon.

Listaus 7.3: Alkuperäisen *recording*-haun kysely.

```
query=london calling \(\live\) AND artist:"the clash"
```

Kun kokeilemme ottaa pois listauksen 7.2 kyselystä kaikki ensimmäisestä OR-operaattorista alkaen viimeiseen sulkeeseen asti, siten että jäljellejääneen alkuperäisen kappaleen ympärille on yhdet sulkeet. Tämän toimenpiteen jälkeen muokatulla kyselyllä saadaan sama määrä tuloksia (77 kappaletta) kuin alkuperäisellä, ja myös kappaleet ovat identtisiä. Listauksien 7.1 ja 7.2 kyselyiden yhteydessä tehdyt kokeilut osoittavat, että sulkeilla kappaleen ympärillä on suuri vaikutus saatuihin tuloksiin. Kummallekin esimerkkikappaleelle – *Lily, Rosemary And The Jack o, London Calling (live)* – saadaan oikea julkaisu ympäröimällä ne vain sulkeilla. Apuna ei tarvitse käyttää jokerimerkkiä eikä OR-operaattoria.

Viimeinen haku tehdään käyttäen pelkkää artistin nimeä, jos tuloksia ei olla edellisillä saatu. Tämä vaikuttaa turhalla, koska artisteilla voi olla valtava määrä kappaleita, ja vaikka tuloksissa sattuisi olemaan kappale kuunneltavalta albumilta, hyvin suurella todennäköisyydellä se ei ole kappale, jota kuunnellaan. Suoritetun analyysin perusteella voidaan sanoa, että prototyypissä tehdään turhia *recording*-hakuja ja ensimmäinen kysely ei ole täysin oikein muodostettu ja jokerimerkin käyttö on tarpeetonta. Näyttäisi myös siltä, että yksi lisähaku on riittävä, jos ensimmäinen haku ei tuota tuloksia. Haut ovat tällöin seuraavanlaiset:

1. query=(*kappaleen nimi*) AND artist:"*artistin nimi*"
2. query=(*kappaleen nimi*)

Haettavien kappaleiden maksimimäärä. MusicBrainzista on mahdollista hakea yhdellä pyynnöllä maksimissaan sadan kappaleen tiedot. Usein sitä enempää tuloksia ei tule, mutta suosittujen ja pitkän uran tehneiden artistien yhteydessä saattaa tuloksia tulla huomattavasti enemmän, mitä yksi pyyntö voi palauttaa. Esimerkiksi listauksen 7.1 hakupyynnöllä saadaan 2786 kappaletta.

Prototyyppi tallentaa vain 100 ensimmäisen kappaleen sisältämät julkaisut muistiin. Yleensä oikea julkaisu on myös näiden 100 ensimmäisen joukossa, mutta poikkeuksiakin on. Esimerkiksi jos kuunnellaan Nirvanan kappaletta *Come as You Are* kokoelma-albumilta *Nirvana*, *recording*-haku on tällöin listauksen 7.4 mukainen.

Listaus 7.4: Haun *recording* kysely artistin *Nirvana* kappaleen *Come as You Are* tietojen hakemiseen.

```
query=(come as you are) AND artist:"nirvana"
```

Haun palauttamien 100 ensimmäisen kappaleen joukossa ei ole kokoelma-albumilla *Nirvana* olevaa kappaletta. Tuloksia saatiin yhteensä 876, joiden joukossa halutun julkaisun kappale on sijalla 301.

Vaikuttaa siltä, että prototyypin toiminnan kannalta on järkevää hakea enemmän kuin vain 100 ensimmäistä. Mutta jos aina haetaan kaikki tulokset, MusicBrainziin tehtävien pyyntöjen määrä saattaa kasvaa melko suureksi. Listauksen 7.1 tapauksessa Bob Dylanin kappaleelle "*Lily, Rosemary And The Jack o*" pyyntöjä jouduttaisiin tekemään yhteensä 28. Ongelmaksi tämä voisi muodostua esimerkiksi kuunneltaessa lyhyttä kappaletta, jolle saadaan suuri määrä tuloksi. Kappaleen albumi voi jäädä tällöin tunnistamatta, koska kaikkia pyyntöjä ei ehditä tehdä. Jos kappale on kuuntelun tunnistamisen kannalta kriittisellä kohdalla eli albumin lopussa, ei albumia pystyittäisi tunnistamaan kuunnelluksi.

Tulosjoukon rajaaminen. Kyselyitä on mahdollista rajata *recording*-haussa erilaisten attribuuttien avulla. Kuunteluntunnistajan kannalta hyödyllisimpiä ovat julkaisuun liittyvät attribuutit. Niiden avulla voidaan hakea esimerkiksi vain julkaisut, joiden status on *official*. Tämä rajaa epäviralliset *bootleg*-julkaisut pois tuloksista. Myös julkaisun tyyppillä (*album*, *compilation*, *live*,...) voidaan kaventaa tulosjoukkoa. Rajaustavoista käyttökelpoisin on statuksella rajaaminen, koska julkaisun tyyppi rajaa liian voimakkaasti tulosjoukkoa.

Rajaamalla tulosjoukko sisältämään ainoastaan kappaleet virallisilta julkaisuilta, saadaan haluttu julkaisu suurella todennäköisyydellä mahtumaan ensimmäisen hakukyselyn palauttamiin tuloksiin. Jos haluttu julkaisu ei mahdu ensimmäisten joukkoon, voidaan käyttää sivutusta, jonka avulla loput tulokset saadaan haettua. Listauksen 7.4 kyselyn tapauksessa Nirvanan kappaleelle *Come as You Are* saatavien tulosten määrä pienenee merkittävästi 876 kappaleesta 44 kappaleeseen.

Tulosjoukon rajaaminen statuksella *official* näyttäisi olevan jo yksinään pätevä ratkaisu halutun julkaisun sisällyttämiseen ensimmäisen pyynnön palauttamaan tulosjoukkoon. Haittapuolena on, että jotkin harvinaiset julkaisut jäävät tunnistamatta.

7.1.2 Oikean julkaisun valinta

Kuunneltavan albumin päätteleminen. Prototyyppi ei toteuta täysin kohdassa 2.2 esitettyä albumin tunnistuksen teoriaa, jossa vertailujulkaisun valinnassa hyödynnettiin kappaleiden järjestysnumeroita. Kuunneltavan albumin päättele-

minen pohjautuu kuuntelupuskuriin, joka pitää yllä tietyn mittaista kuunteluhistoriaa. Teoriaa ei pystytä noudattamaan, koska kuunneltavan albumin tiedot eivät aina ole yhdenmukaisia metatietokannan kanssa. Tämän vuoksi päädyttiin joustavampaan mutta ei aivan yhtä tarkkaan tapaan, joka on esitetty alakohdassa 6.2.3. Julkaisuksi valitaan se, joka esiintyy eniten kuuntelupuskurissa ja on useimmiten hakutulosten kärjessä. Tällä tavalla saadaan kuunneltava albumi suurella todennäköisyydellä, mutta ei välttämättä oikeaa julkaisua. Julkaisuksi voi tulla esimerkiksi bonusjulkaisu, joka sisältää ylimääräisiä kappaleita. Tästä syystä kuunneltavaa albumia ei aina pystytä tunnistamaan, koska kuunteluntunnistaja luulee, että albumia ei olla kuunneltu kokonaan.

Hakutulossijoitusten lisäksi kuuntelupuskuri järjestää julkaisut eräiden julkaisukohtaisten attribuuttien mukaan. Tämä tehdään, jotta albumin julkaisuksi tulisi todennäköisemmin oikea. Attribuutit ovat julkaisun tyyppi (*type*) ja julkaisuvuosi (*date*). Sellaiset julkaisut, joiden tyyppi on *album* nousevat korkeammalle kuin muun tyyppiset julkaisut. Myös sellaisia julkaisuja suositaan joilla on julkaisuvuosi.

Yksi vaihtoehto kuunneltavan albumin tunnistamisen parantamiseksi voisi olla suosia julkaisuja, jotka sisältävät pienemmän kappalemäärän. Tämän ansiosta albumi voitaisiin tunnistaa riittävän ajoissa. Haittapuolena on, että albumi saatetaan tunnistaa kuunnelluksi useampaan kertaan, koska albumi voi sisältää enemmän kappaleita kuin vertailujulkaisu. Ensin tunnistetaan vähemmän kappaleita sisältävä vertailujulkaisu. Tämän jälkeen vertailujulkaisuksi tulee kuunneltavan albumin julkaisu, joka tunnistetaan myös kuunnelluksi.

Kuuntelupuskurin koko. Kuuntelupuskurin ylläpitämällä kappalehistorian pituudella on suuri vaikutus albumin tunnistuksen tarkkuuteen. Liian lyhyellä puskurilla vertailujulkaisu saattaa vaihdella kuuntelun edetessä, eikä tällöin välttämättä ole oikea julkaisu valittuna albumikuuntelun tunnistamisen kannalta kriittisellä hetkellä eli albumin lopussa. Liian pitkä puskuuri voi puolestaan haitata albumin tunnistusta. Ongelma saattaa tulla esiin, kun kuunnellaan tietyltä artistilta kappaleita ensin satunnaisesti ja sitten järjestyksessä sellaiselta albumilta, jolla on vähän kappaleita. Tällöin vertailujulkaisu ei mahdollisesti ehdi vaihtua albumiksi, jota kuunnellaan. Kuuntelupuskurin koon suuruus ei olisi ongelma, jos puskuuri pystyttäisiin nollaamaan aina albumin vaihtuessa. Soittimelta ei kuitenkaan saada aina albumin nimeä, koska kohdassa 2.1 käsiteltyjen rajapintojen tarjoama tiedon määrä on rajallinen. Tästä syystä kuuntelupuskuria ei voida pitää kiinteästi suurena.

Prototyypissä kuuntelupuskurin koko on vakiintunut kolmen kappaleen mittaiseksi. Sen kokoisella puskurilla pystytään reagoimaan riittävän nopeasti muutoksiin ja se tunnistaa myös hyvin kuunneltavan albumin. Kolmen kappaleen mittaiselle puskurille ongelmallisiksi ovat osoittautuneet sellaisten artistien albumit, jotka sisältä-

vät paljon samoista kappaleista koostuvia albumeja. Tällaisia albumeja ovat varsinkin kokoelma-julkaisut. Tämän vuoksi kuuntelupuskurissa olevat kolme kappaletta saattavat esiintyä samanaikaisesti monella julkaisulla. Esimerkiksi jos kuuntelupuskurin kokoa pystyttäisiin kasvattamaan dynaamisesti kokoelmajulkaisuja kuunneltaessa, ei muulloin lyhyt puskuri olisi ongelma.

7.1.3 Kappalelistan hakeminen ja kuuntelun tunnistus

Prototyyppi hyödyntää *release*-haun avulla saatavaa julkaisun kappalelistaa kuuntelun tunnistamisessa. Kahta viimeksi kuunneltua kappaletta verrataan tähän listaan. Albumi ollaan kuunneltu, jos kappaleet täsmäävät kahteen toiseksi viimeiseen listalla olevaan kappaleeseen. Kappalelistaan vertaaminen vaikuttaa tämän hetkisen toteutuksen kannalta turhalta, koska kuuntelupuskurista saadaan suoraan riittävät tiedot kuuntelun tunnistamiseen.

Albumikuuntelun tunnistus ilman kappalelistaa onnistuu vertaamalla kuuntelupuskurin kahta viimeistä kappaletta keskenään. Vertailussa käytetään suosituimman julkaisun kappaleita. Albumi voidaan todeta kuunnelluksi jos kappaleet ovat peräkkäiset ja puskurin viimeinen kappale on toiseksi viimeisenä julkaisulla. Tämä kahden toiseksi viimeisen kappaleen perusteella tehtävä albumikuuntelun tunnistus tuo joustavuutta, sillä albumin alku ja keskivaiheilla tapahtuneet kappaleiden tunnistusvirheet eivät haittaa kuuntelun tunnistusta. Haittapuolena tässä ratkaisussa ovat virheellisesti kuunnelluksi tulevat albumit. Jos esimerkiksi kuunnellaan vain muutama kappale albumin lopusta, albumia ei tällöin olla kuunneltu kokonaan, eikä kuuntelua tällöin tulisi lähettää Statsteriin.

Vertailujulkaisun kappalelistan hakeminen *release*-haulla ei ole aivan välttämätöntä, koska kuuntelun tunnistus on mahdollista tehdä kuuntelupuskurin avulla. Vain yhdestä mediasta koostuvat albumit on mahdollista tunnistaa kuunnelluiksi käyttäen pelkästään kuuntelupuskuria. Useammasta mediasta koostuvat albumit eivät enää ole yhtä helposti tunnistettavissa käyttäen ainoastaan kuuntelupuskurista saatavia tietoja. Useista medioista koostuvien julkaisujen kuuntelun tunnistamiseen tarvitaan medioiden lukumäärä. Tämä tieto saadaan vain *release*-haulla. Sen avulla saadaan myös tietoa, josta voi olla tulevaisuudessa hyötyä, kuten median formaatti (*format*) ja yksittäisen median otsikkoteksti (*title*). Median formaattia voidaan esimerkiksi käyttää vertailujulkaisun valinnassa apuna, koska tietokoneella kuunneltava musiikki on hyvin todennäköisesti CD-levyltä. Tämän vuoksi voidaan CD-levyformaattillisia julkaisuja suosia vertailujulkaisun valinnassa.

7.2 Tunnetut puutteet

Prototyypitoteutus ei ole vielä aivan täydellinen. Kuuntelun tunnistamisessa ilmenee ongelmia joissain tapauksissa. Neljä selkeää ongelmakohtaa voidaan nostaa esiin: kokoelmajulkaisut, kuunnelluksi tunnistaminen kahden kappaleen perusteella, bonuskappaleita sisältävät julkaisut ja albumit, joita ei ole MusicBrainzissa. Näistä kaksi ensimmäistä on käsitelty alakohdissa 7.1.2 ja 7.1.3.

Bonusjulkaisujen yhteydessä esiintyvä puute aiheuttaa sen, että albumi tulee joko tunnistetuksi kahteen kertaan tai sitä ei pystytä tunnistamaan kuunnelluksi lainkaan. Moninkertainen tunnistus johtuu siitä, että ensin kuunnelluksi tunnistetaan normaali julkaisu ja tämän jälkeen bonusjulkaisu. Albumi tulisi tunnistaa vain kerran kuunnelluksi joko alkuperäiseksi tai bonusjulkaisuksi. Kokonaan tunnistamatta jääminen ilmenee normaalien julkaisujen yhteydessä. Prototyyppi saattaa virheellisesti tunnistaa kuunneltavan albumin bonusjulkaisuksi. Tällaisessa tapauksessa kuuntelun tunnistaja luulee, ettei albumia kuunnella koskaan loppuun, koska kuunneltavan albumin julkaisussa on vähemmän kappaleita kuin vertailujulkaisuna käytettävässä. Tämä ongelma on helposti korjattavissa heti kun bonusjulkaisut pystytään tunnistamaan vain kerran kuunnelluksi.

MusicBrainz on ulkoinen metatietokanta, joten se ei välttämättä sisällä kaikkia käyttäjällä olevia albumeita. Esimerkiksi jotkin uudet kotimaiset albumijulkaisut ovat tällaisia, kuten tässä työssä esimerkkinä jo käytetty artistin *Penniless* albumi *A Cap to the City*. Ratkaisu tähän puutteeseen on joko lisätä albumi MusicBrainziin tai ottaa käyttöön rinnalle Last.fm-palvelun tarjoama metatietokanta.

7.3 Jatkokehitysideoita

Prototyyppiä toteutettaessa tuli esiin useita jatkokehitysideoita liittyen prototyyppiin ja Statster-verkkopalveluun. Ilman verkkopalveluun tehtäviä muutoksia kaikki prototyypin kehitysideat eivät olisi edes mahdollisia.

Prototyypin kehitysideat koskevat sekä toimintavarmuuden lisäämistä että käytettävyyden parantamista. Statsterin kehitysideoilla puolestaan pyritään tuomaan verkkopalvelua lähemmäksi todellisuutta, kuten tapoja yksilöidä paremmin tietyn albumin erityyppisiä julkaisuja.

7.3.1 Statster

Albumin yksilöiminen. Statsterin tietomalli mahdollistaa albumin yksilöimisen ainoastaan nimen avulla. Tästä syntyy ongelma, koska tietyllä artistilla voi olla useampi samanniminen albumi. Esimerkiksi artistilla *Killswitch Engage* on kaksi *Killswitch Engage* -nimistä albumia. Albumeista ensimmäinen on julkaistu vuonna

2000 ja jälkimmäinen 2009. Statsterissa albumit on eritelty nimen avulla, vanhin albumi on nimellä *Killswitch Engage* ja uudempi *Killswitch Engage II*. Prototyypillä tunnistettaessa kuuntelu lisätään molempien albumien tapauksessa aina *Killswitch Engage* -nimiselle albumille, koska Statster ei huomioi julkaisuvuotta albumin yksilöimisessä.

Statsteriin on toteutettu komponentti nimeltä yhtenäistäjä, jolla väärin nimetyt albumikuunteluita voidaan ohjata oikeille albumeille. Albumien yksilöimisessä käytetään artistin ja albumin nimeä. Albumin *Killswitch Engage* (2009) kuuntelut voitaisiin ohjata yhtenäistäjän avulla albumille *Killswitch Engage II*, jos se hyödyntäisi myös julkaisuvuotta albumien yksilöimisessä. Tämä muutos on suhteellisen helppo toteuttaa.

MusicBrainzin tietomallin hyödyntäminen Statsterissa voisi olla toinen keino albumikuunteluiden yksilöimiseen ja eri tyyppisten julkaisujen ryhmittelyyn. MusicBrainz kokoaa tietystä albumista tehdyt julkaisut *release-group* nimiseksi kokonaisuudeksi, joka vastaa käsitteellisesti Statsterin albumia. Statsterin tietomallia pitäisi muuttaa vähintään siten, että myös julkaisuvuosi olisi yksi albumikuuntelun yksilöivistä attribuuteista. Tämä on jo kuitenkin huomattavasti suurempi muutos verrattuna yhtenäistäjän päivittämiseen.

Palvelurajapinnan laajentaminen. Statsterin palvelurajapinta mahdollistaa vain albumikuuntelujen lisäämisen ja kuunteluvirran hakemisen. Rajapinnan avulla palveluun olisi hyvä pystyä olemaan paremmin vuorovaikutuksessa, jotta palvelu pystyisi luomaan ympärilleen enemmän käyttäjiä hyödyttäviä oheisohjelmia.

Hyödyllinen lisä rajapintaan olisi albumien ja artistien suosikkeihin lisäysmahdollisuus, jotta albumikuuntelun tunnistajan avulla voitaisiin tykätä kuunnelluista albumeista ja fanittaa artisteja. Myös tietyn käyttäjän suosikkien hakeminen olisi hyvä lisä. Tämän avulla voitaisiin esimerkiksi luoda soittolista, joka koostuu käyttäjän tykkäämistä albumeista.

7.3.2 Prototyyppi

Bonusjulkaisujen tunnistus. Albumikuuntelun tunnistaja ei käyttäydy toivottulla tavalla bonusjulkaisuja kuunneltaessa. Bonusjulkaisut tunnistetaan usein kahteen kertaan kuunnelluiksi. Tätä käsiteltiin tarkemmin kohdassa 7.2.

Tämä puute on yksi tärkeimmistä jatkokehitystä vaativista asioista. Ilman sen korjaamista käyttäjän kuunteluhistoria saattaisi vääristyä. Kyseisen puutteen vuoksi albuminkuuntelun tunnistajaa ei tulisi edes antaa yleiseen käyttöön.

Kuuntelupuskuri. Toinen jatkokehitystä vaativa osa albumikuuntelun tunnistajassa on kuuntelupuskuri. Dynaaminen kuuntelupuskuri ja vertailujulkaisun valin-

nan hiominen ovat tärkeitä kehityskohteita.

Dynaaminen kuuntelupuskuri mukautuu kasvattamalla kokoaan kokoelma-albumeita kuunneltaessa ja pienentymällä muiden albumien yhteydessä. Arvoväli voisi olla esimerkiksi 3–10. Yläraja on tärkeä, jotta puskurin pituus pysyisi aina järkevän mittaisena.

Vertailujulkaisun valinnassa on jonkin verran kehitettävää. Julkaisuista on saatavilla paljon erilaista tietoa, joita hyödyntäen albumin tunnistusta voitaisiin tarkentaa. Julkaisun sisältämien kappaleiden määrä on yksi tällainen ominaisuus ja myös julkaisun tyyppi.

Last.fm-palvelun hyödyntäminen. Albumikuuntelun tunnistusta voidaan parantaa ottamalla rinnalle toinen metatietokanta. Last.fm-palvelun tarjoama metatietokanta on saatavilla olevista varteenotettavin. Sen avulla pystyttäisiin tunnistamaan suuri osa albumeista, joita MusicBrainzissa ei ole.

Korjauslistat. Kappaleen kirjoitusasun korjauslista olisi jossain tapauksissa hyödyllinen. Korjauslista voisi olla yleinen tai käyttäjän määritettävissä. Esimerkiksi jos kuunnellaan artistin *Pink Floyd* albumia *Wish You Were Here*, ja albumilta kappaletta “*Shine On You Crazy Diamond (part one)*”, joka on eritavalla kirjoitettu kuin MusicBrainzissa oleva nimi “*Shine On You Crazy Diamond, Parts I-V*”. Tästä syystä haluttua albumia ei löydy. Jos albumikuuntelun tunnistajalle pystyttäisiin kertomaan, että kuunneltava kappale ei vastaa MusicBrainzissa olevaa kappaleen nimeä. Tällöin se voisi käyttää korjauslistassa olevaa vastinetta tietojen hakemiseen, ja oikea julkaisu löytyisi.

8. JOHTOPÄÄTÖKSET

Tässä diplomityössä perehdyttiin soitinohjelmistosta riippumattomaan tapaan tunnistaa albumikuuntelu. Albumikuuntelun tunnistuksen teoria koostui kahdesta kokonaisuudesta: soittimien ulkoisista rajapinnoista ja metatietokannoista. Rajapintoihin tutustuttiin ainoastaan teoriassa. Metatietokantoja puolestaan tarkasteltiin teorian lisäksi käytännössä. Työtä varten toteutettiin prototyypisovellus, joka hyödynsi verkkopalvelun MusicBrainz tarjoamaa metatietokantaa albumikuuntelun tunnistuksessa.

Soittimien tarjoamiin rajapintoihin perehtyminen osoitti, että kuunneltavan kappaleen tiedot on mahdollista saada lähestulkoon jokaiselta soittimelta joutumatta tekemään muutoksia itse soitinohjelmistoon. Jos soittimeen ei kuitenkaan ole toteutettu integroitua rajapintaa, on olemassa muita keinoja saada kuunneltavan kappaleen tiedot. Soittimiin on usein tehty lisäosia, joiden avulla kuunneltavan kappaleen tiedot lähetttävä rajapinta saadaan otettua käyttöön. Esimerkiksi MSN Messenger -rajapinnan toteuttavia lisäosia on hyvin saatavilla. Kuunneltavien kappaleiden tietojen saaminen soittimilta ei ole este itsenäisen albumikuuntelun tunnistajan toteuttamiselle.

Metatietokantojen teoriaosa nosti esille verkkopalvelut Last.fm ja MusicBrainz, jotka tarjoavat käyttökelpoisen metatietokannan albumikuuntelun tunnistamiseen. Teorian pohjalta valittiin MusicBrainz prototyypitoteutuksen metatietokannaksi muun muassa sen laajan palvelurajapinnan vuoksi. Prototyyppi osoitti, että albumikuuntelun tunnistus on mahdollista soittimesta riippumattomalla sovelluksella, kun saatavilla on ainoastaan kuunneltavan kappaleen nimi ja artisti. Nämä tiedot saadaan soittimilta työssä tutkittujen rajapintojen avulla.

Työ osoitti, että automaattinen albumikuuntelun tunnistus on mahdollista toteuttaa erilleen soitinohjelmistosta omaksi itsenäiseksi sovellukseksi. Tämän vuoksi jokaiselle soittimelle ei tarvitse kirjoittaa omaa ohjelmaa. Käyttäjille on mahdollista tarjota kohtalaisella vaivalla ohjelmisto, joka huolehtii albumikuunteluiden tunnistamisesta ja lähettämisestä soitinohjelmistosta riippumatta.

Prototyypistä on kuitenkin vielä matkaa itsenäiseen albumikuuntelun tunnistajaan. On olemassa joitain tunnistamisen luotettavuuteen liittyviä ongelmia, jotka on ratkaistava ennen kuin voidaan tehdä itsenäinen, soittimien ulkoisia rajapintoja hyödyntävä sovellusta.

LÄHTEET

- Apache Lucene - Query Parser Syntax.* (2011, Maaliskuu). Verkkosivu. The Apache Software Foundation. Viitattu 10.05.2011, saatavilla http://lucene.apache.org/java/3_1_0/queryparsersyntax.html
- Audioscrobbler Plugins.* (s.a.). Last.fm. Viitattu 21.04.2011, saatavilla <http://build.last.fm/category/Scrobblers>
- Büscher, W. (2004, 10). *Communication between windows programs using WM_COPYDATA messages.* Viitattu 10.04.2011, saatavilla http://www.qsl.net/dl4yhf/yhf_comm/yhf_comm_info.htm
- Discogs API Documentation.* (s.a.). Verkkosivu. Viitattu 11.05.2011, saatavilla <http://www.discogs.com/help/api>
- Extensible Markup Language (XML) 1.0 (Fifth Edition).* (2008, Marraskuu). Verkkosivu. Viitattu 21.07.2011, saatavilla <http://www.w3.org/TR/REC-xml/>
- Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures.* Väitöskirja, University of California, Irvine, CA. Viitattu 23.06.2011, saatavilla <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- foobar2000.* (2011, Toukokuu). Wiki-sivusto. Viitattu 26.05.2011, saatavilla <http://wiki.hydrogenaudio.org/index.php?title=Foobar2000:Foobar2000&oldid=22781>
- Foobar2000:Title Formatting Reference.* (2010, Lokakuu). Wiki-sivu. Viitattu 30.05.2011, saatavilla http://wiki.hydrogenaudio.org/index.php?title=Foobar2000:Title_Formatting_Reference&oldid=22028
- Guy, T. (2010, Syyskuu). *VBScript - To Check your Versions.* Verkkosivu. Viitattu 07.06.2011, saatavilla http://www.computerperformance.co.uk/Logon/VBScript/VBScript_Windows_version.htm
- HOW TO: synchronise your last.fm playback stats with foobar.* (s.a.). Forum post. Viitattu 27.05.2011, saatavilla <http://www.hydrogenaudio.org/forums/index.php?showtopic=76772>
- Hypertext Transfer Protocol – HTTP/1.1.* (1999, Kesäkuu). Verkkosivu. Viitattu 25.07.2011, saatavilla <http://www.ietf.org/rfc/rfc2616.txt>

- Kesteren, A. van. (2010, Elokuu). *XMLHttpRequest*. Dokumentaationsivu. Viitattu 29.05.2011, saatavilla <http://www.w3.org/TR/2010/CR-XMLHttpRequest-20100803/>
- Last.fm statistics*. (2008, Helmikuu). Viitattu 06.05.2011, saatavilla <http://socialmediastatistics.wikidot.com/lastfm>
- Last.fm Web Services*. (s.a.). Verkkosivu. Viitattu 12.05.2011, saatavilla <http://www.last.fm/api>
- Libre.fm About*. (s.a.). Verkkosivu. Viitattu 18.07.2011, saatavilla <http://libre.fm/about/>
- Light of Day, Day of Darkness*. (s.a.). Verkkosivu. Viitattu 14.06.2011, saatavilla <http://www.allmusic.com/album/r565098>
- Mathews, L. (2011, Tammikuu). *Ubuntu 11.04 officially switches default media player to Banshee*. Netti-sivu. Viitattu 04.05.2011, saatavilla <http://downloadsquad.switched.com/2011/01/09/ubuntu-11-04-officially-switches-default-media-player-to-banshee/>
- MPRIS D-Bus Interface Specification*. (2008). Dokumentaationsivu. Viitattu 16.06.2011, saatavilla <http://www.mpris.org/1.0/spec.html>
- MPRIS v1.0 Metadata guidelines*. (s.a.). Wiki-sivu. Viitattu 16.06.2011, saatavilla http://xmms2.org/wiki/MPRIS_Metadata#MPRIS_v1.0_Metadata_guidelines
- MSN Messenger Now Playing Plugin*. (s.a.). Verkkosivu. Viitattu 18.07.2011, saatavilla <http://www.myplugins.info/msn-messenger-plugin.php>
- MusicBrainz database statistics*. (2011, Toukokuu). Verkkosivu. MusicBrainz. Viitattu 06.05.2011, saatavilla <http://musicbrainz.org/show/stats/>
- MusicBrainz History*. (s.a.). Verkkosivu. Viitattu 17.06.2011, saatavilla http://musicbrainz.org/doc/MusicBrainz_History
- MusicBrainz Timeline – Next Generation Schema*. (s.a.). Verkkosivu. Viitattu 17.06.2011, saatavilla <http://musicbrainz.org/doc/Timeline>
- OSI model*. (s.a.). Wiki-sivu. Viitattu 25.07.2011, saatavilla http://en.wikipedia.org/w/index.php?title=OSI_model&oldid=439016354
- Sansome, D. (s.a.). *Controlling Clementine from the commandline with DBus and MPRIS*. Viitattu 16.06.2011, saatavilla <http://code.google.com/p/clementine-player/wiki/MPRIS>

- Schoenmakers, B. (2009, Huhtikuu). *Control KDE applications from the commandline with D-Bus*. Verkkosivu. Viitattu 18.07.2011, saatavilla <http://www.bramsschoenmakers.nl/en/node/609>
- Scrobbles*. (s.a.). Verkkosivu. Viitattu 15.08.2011, saatavilla <http://build.last.fm/category/Scrobbles>
- Statster – lisäosat*. (s.a.). Verkkosivu. Viitattu 13.07.2011, saatavilla <http://statster.info/plugins/>
- Top Ten Distributions*. (s.a.). Verkkosivu. Viitattu 18.07.2011, saatavilla <http://distrowatch.com/dwres.php?resource=major&language=EN>
- Turney, J. (s.a.). *How do I configure my player and musictracker to work together?* Verkkosivu. Viitattu 11.04.2011, saatavilla <http://code.google.com/p/pidgin-musictracker/wiki/Configuring>
- Wang, T. (2009, Helmikuu). *WSH Panel Mod*. Forum post. Viitattu 25.05.2011, saatavilla <http://www.hydrogenaudio.org/forums/index.php?showtopic=70363>
- What is a Bootleg*. (s.a.). Verkkosivu. Viitattu 19.07.2011, saatavilla <http://www.bobsboots.com/boot.html>
- What is freedb?* (s.a.). Verkkosivu. Viitattu 13.07.2011, saatavilla <http://www.freedb.org/en/faq.3.html#11>
- What is ID3 (v1)?* (s.a.). Verkkosivu. Viitattu 20.07.2011, saatavilla <http://www.id3.org/ID3v1>
- What Is WSH?* (2009, Maaliskuu). Dokumentaationsivu. Viitattu 07.06.2011, saatavilla [http://msdn.microsoft.com/en-us/library/shzd7dy4\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/shzd7dy4(VS.85).aspx) (MSDN)
- Which media players does the Last.fm Scrobbler support?* (s.a.). Verkkosivu. Viitattu 20.07.2011, saatavilla <http://www.last.fm/help/faq?category=100#180>
- Winer, D. (1999, Kesäkuu). *XML-RPC Specification*. Verkkosivu. Viitattu 25.07.2011, saatavilla <http://www.xmlrpc.com/spec>
- XML Web Service*. (s.a.). Wiki-sivu. MusicBrainz. Viitattu 11.05.2011, saatavilla <http://wiki.musicbrainz.org/XMLWebService>